Alchemy

the ultimate sample manipulation synthesiser

camel audio

Table of Contents

1 - Welcome to Alchemy....6
1.1 - Using the Manual....7
1.2 - Manual Conventions....7
1.3 - Downloadable PDF Manual....7
2 - Overview....8
2.1 - Preset Browser Mode....8
2.2 - Simple Mode....9
2. - The Performance section....9
2.3 - Advanced Mode....10
2.4 - Signal flow in Alchemy....11
3 - Using the Interface....12
3.1 - Working with knobs....12
3.2 - Working with buttons and value fields....12
3.3 - MIDI learn....13
4 - Title Bar....14
4.1 - Parameter value display....14
4.2 - Browse, Simple and Advanced buttons....14
4.3 - File, Save and Saveas buttons....14
4.4 - Display options....15
4.5 - Other File menu options....15
4.6 - Random button....15
4.7 - Volume knob....15
4.8 - Advanced: Creating your own randomisation templates....15
5 - Performance Section....17
5.1 - Perform Controls....17
5.2 - Remix pad....17
5.3 - Auto-assign and other controls....18
5.4 - Remix pad keyswitching....18
5.5 - Modwheel....18
5.6 - Recommended perform control assignments....18
6 - Preset Browser....19
6.1 - Loading presets....19
6.2 - The browser results list....19
6.2.1 - Sorting the results list....19
6.3 - Attribute columns....19
6.3.1 - Additional attribute columns....20
6.4 - Rating presets....20
6.5 - Preset properties....20
6.6 - Text searching....20
6.6.1 - Wild card searches....20
6.6.2 - Negated search terms....20
6.7 - Selecting presets using MIDI Bank and Program Change messages....21
6.8 - Edit mode....22
6.8.1 - Applying New Attributes, User Tags and Comments....22
6.9 - Backup or transfer preset ratings, comments and tags....23
7 - The Arpeggiator....24
7.1 - Basic controls....24
7.2 - Using the Arp Sequencer step editors....25
7.3 - Importing from a MIDI file....26
7.4 - Arpeggiator examples....26
7.4.1 - Creating classic arpeggiator patterns....26
7.4.2 - Example: Creating a step sequencer....27
7.4.3 - Make it stop!!!....27
8 - Effects Rack....28
8.1 - Global page controls....28
8.2 - Controlling individual effects....28
8.3 - Effect examples....31

8.3.1 - Ping Pong Delay and Flanger....31
8.3.2 - Multi-band Distortion using Bandpass Filter and Band Reject....32
9 - Sources....33
9.1 - Source elements....33
9.1.1 - Source filters....33
9.1.2 - FilterMix....34
9.1.3 - Global page controls....34
9.1.4 - Source master controls....35
9.1.5 - Source sub-page controls....35
9.1.6 - Modulating Position....36
9.1.7 - Elements and filters....37
9.1.8 - Example: Tempo-synced loops....37
9.2 - Morphing....39
9.2.1 - What parameters participate in a morph?....40
9.2.2 - Where did the 'Morph Square' go?....40
9.2.3 - Morphing Examples....41
9.3 - Importing Audio....43
9.3.1 - Importing multiple files....43
9.3.2 - Granular Import....43
9.3.3 - Additive Import....44
9.3.4 - Add+Spec Import....44
9.3.5 - Spectral Import....44
9.3.6 - 'Import Audio' versus 'Load Audio'....44
9.3.7 - Supported SFZ opcodes....45
9.4 - A Limit on the Size of Imported Sample Data....45
9.5 - A Limit on the Size of Sample Folders....46
10 - Additive Synthesis....47
10.1 - A brief introduction to additive synthesis....47
10.2 - ADD and VA mode buttons....47
10.3 - Using the Additive sub-page controls in ADD mode....47
10.4 - Using the Additive Formant Filter....49
10.4.1 - Example: Manipulating a resynthesised voice....49
10.5 - Advanced: Creating your own Pitch, Amp, and Pan profiles....50
11 - Virtual Analog (VA)....52
11.1 - VA techniques in Alchemy....52
11.1.1 - Example: Creating a 'supersaw' effect....52
12 - Spectral Synthesis....54
12.1 - A brief introduction to spectral synthesis....54
12.2 - Modes of operation....54
12.3 - Controls....54
12.4 - Using the Spectral Formant Filter....54
13 - Granular Resynthesis....56
13.1 - A brief introduction to granular synthesis....56
13.2 - GRANULAR and SAMPLER mode buttons....56
13.3 - Granular sub-page controls....56
13.4 - Using Sampler mode....57
14 - Source Editor....58
14.1 - Using the Main Editor....59
14.1.1 - What do warp markers do?....59
14.1.2 - Warp marker tips....60
15 - Additive Editor....61
15.1 - Overview....61
15.2 - Breakpoints and partial bars....61
15.3 - Detail knob....63
15.4 - Partial bars display and controls....64
15.5 - Breakpoint envelope display and controls....66
15.6 - Image import....67
15.7 - Example: Creating a simple additive sound from scratch....67
16 - Spectral Editor....69
16.1 - Overview....69

16.2 - Spectral editor controls....69
16.3 - Spectral canvas....70
16.4 - Example: Modifying a drum loop....71
17 - Main Filters....72
17.1 - Global page controls....73
17.2 - Filter VU meters....73
17.3 - Filters and filter types....73
17.3.1 - Filter types: lowpass, bandpass, highpass....73
17.3.2 - Filter types: formant and peaking....74
17.3.3 - Filter types: comb filters....74
17.3.4 - Filter types: ring modulation....75
17.3.5 - Filter types: distortion....75
17.4 - Example: Creating vowel sounds with parallel source filters....75
18 - Master Section....77
18.1 - Master voice controls....77
18.2 - Other Master controls....77
18.3 - The PRESERVE button....78
19 - Modulation....79
19.1 - Working with modulators....79
19.2 - Types of modulator....80
19.3 - Modulation rack controls....80
19.3.1 - Modulator knobs are different....80
19.3.2 - Finding a modulator's targets....81
19.3.3 - A very simple modulation example....81
19.3.4 - Modulating LFO Rate....81
19.4 - LFO....82
19.5 - AHDSR envelopes....83
19.5.1 - Envelope shaping....83
19.6 - MSEG....83
19.6.1 - Editing MSEG envelopes....84
19.6.2 - Edit modes....84
19.6.3 - Sync Mode....84
19.6.4 - Loop markers....84
- The Arpeggiator - 7....1
19.6.5 - Other controls and parameters....85
19.6.6 - Example: Modulating pitch....85
19.7 - Sequencer....85
19.7.1 - Importing from a MIDI file....86
19.8 - Note Property....87
19.9 - Perform....88
19.10 - ModMap....88
19.10.1 - Assigning a ModMap....88
19.10.2 - ModMap recipes....89
19.10.3 - The details: how ModMaps work....91
19.11 - XY-MSEG....93
19.11.1 - Using the XY MSEG controls....93
- Using the XY MSEG breakpoint editor and square....94
- A quick illustration....94
- A quick illustration....94
20 - SFZ Files....95
20.1 - What are SFZ files?....95
20.2 - Creating your own SFZ files....95
20.3 - Opcodes recognised by Alchemy....95
21 - Automation Names....97
22 - Additive and spectral data....105
22.1 - Creating your own Additive/Spectral Sounds in CSV Format....105
22.2 - Data Format Description....106
22.3 - Amplitude scaling systems....107
23 - LFO Shapes illustrated....108
23.1 - Basic....108

23.2 - Serial-Angular..... 109
23.3 - Serial-Smooth..... 112
23.4 - UHF..... 114
24 - Custom oscillators and LFOs..... 117
24.1 - Custom LFO waveforms..... 117
24.2 - Custom Oscillator waveforms..... 117
25 - Cameleon to Alchemy..... 118
25.1 - Preset types and terminology..... 119
26 - Troubleshooting..... 120
26.1.1 - There is distortion — how do I get rid of it?..... 120
26.1.2 - The sound is too quiet — how do I make it louder?..... 120
26.1.3 - There is a 'click' at the start of each note — How do I get rid of it?..... 120
26.1.4 - There are 'stuck' notes — how do I eliminate them?..... 120
26.1.5 - When using an MSEG as the 'master amp envelope', ensure that the modulation depth is set to exactly 100%..... 120
26.1.6 - Alchemy's CPU usage is too high - how do I reduce it?..... 120
26.1.7 - A dialog box has appeared saying 'Please select a location and filename for supporting data files…' — what should I do?..... 121
26.1.8 - There are no presets — why is that?..... 121
26.1.9 - Alchemy is asking for a 'keyfile' — what should I do?..... 121
- 7 - The Arpeggiator 27..... 1
27 - Sound-designer Guidelines..... 123
27.1 - Preset Design Criteria..... 123
27.2 - Sample Content Guidelines..... 126
27.2.1 - Sample Content — Special Cases..... 126
27.2.2 - SFZ file format considerations..... 126
28 - Credits..... 128

1 Welcome to Alchemy

Alchemy is the ultimate sample manipulation synthesiser. It's a synth powerhouse and yet is very easy to use thanks to its performance controls, remix pads and preset browser. Just tweak the library of excellent presets from many of the world's top sound designers or dive in and analyse your own samples – it's up to you!

Alchemy features additive, spectral and granular synthesis and resynthesis, sampling, and a very capable virtual analog engine with unison and PWM. You can morph or crossfade between sources. You can import your own samples from SFZ, WAV or AIFF files. A wide range of analog modelled filters is included, in addition to a flexible rack of effects which includes all those from CamelPhat and CamelSpace as well as many new effects such as a high quality reverb. The innovative modulation system is extremely flexible, yet easy to use. Alchemy also features a powerful arpeggiator with the ability to import the groove from any MIDI file for immediate synchronisation to a beat.

Alchemy ships with over 5.5GB of samples and analysed content from in-house designers Tim Conrardy and Biomechanoid, as well as designers such as Ian Boddy, Robert Rich, Scott Solida, Nucleus SoundLab and others. A library of over 650 excellent presets from many of the world's top sound designers is included, arranged into categories and sub-categories with tags and attributes to help you find to the sound you need for your project.

1.1 Using the Manual

A list of topics is available at the beginning of the manual. Click a topic to go directly to its page. You can also click on links found in the main text. Hints and tips appear throughout the manual; pay attention to these because they're designed to help you get the most out of Alchemy.

If this is your first time browsing through the manual, we suggest you begin with the Overview page. Or dive right in to whatever topic interests you most!

1.2 Manual Conventions

Throughout this manual, when referring to the installation files and folders, either the data folder (presets, samples etc.) or the plugin files, the following locations are assumed.

Mac	**Folder**
Configuration folder	`/Library/Application Support/Camel Audio/Alchemy`
Data folder	`/Library/Application Support/Camel Audio/Alchemy`
Audio Unit (AU)	`/Library/Audio/Plug-ins/Components/Alchemy.component`
VST	`/Library/Audio/Plug-ins/VST/Alchemy.vst`
RTAS	`/Library/Application Support/Digidesign/Plug-ins/Alchemy.dpm`
Windows XP / 32-bit	**Folder**
Configuration folder	`Start Menu > All Programs > Camel Audio > Alchemy > Alchemy Configuration`
Data folder	`Start Menu > All Programs > Camel Audio > Alchemy > Alchemy Data`
VST	`C:\Program Files\VSTPlugins`
RTAS	`C:\Program Files\Common Files\Digidesign\DAE\Plugins\Alchemy.dpm`
Windows 7 / 64-bit	**Folder**
Configuration folder	`Start Menu > All Programs > Camel Audio > Alchemy > Alchemy Configuration`
Data folder	`Start Menu > All Programs > Camel Audio > Alchemy > Alchemy Data`
VST (32-bit)	`C:\Program Files (x86)\VSTPlugins`
VST (64-bit)	`C:\Program Files\VSTPlugins`
RTAS (32-bit)	`C:\Program Files (x86)\Common Files\Digidesign\DAE\Plugins\Alchemy.dpm`

1.3 Downloadable PDF Manual

The latest version of this manual is available to download and read offline or print. There is also an online version which has several advantages: it is more easily searchable, has images viewable at different sizes, and is always completely up to date. For a record of the latest changes to the online manual (some of which will not be found in the PDF), see the Manual history page.

2 Overview

Alchemy has three different main views. Each view is accessed by clicking the **Browser**, **Simple** and **Advanced** buttons to the right of the Alchemy logo and version number. The Title Bar and Performance Controls are common to all views.

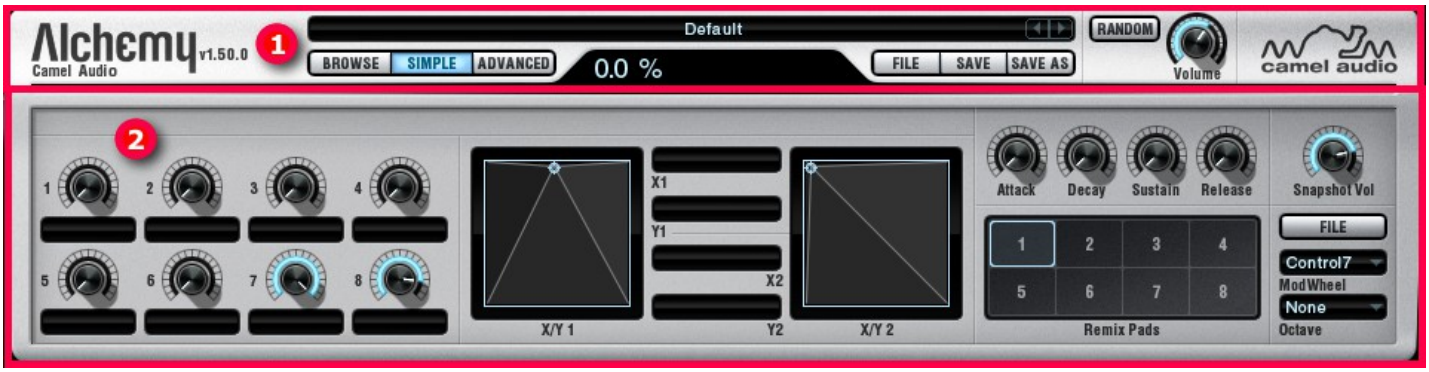
2.1 Preset Browser Mode

The Preset Browser page in Alchemy is divided into five sections, shown above.

1. The Title bar.
2. The browser attributes section.
3. The browser results list.
4. The text search and preset properties section.
5. The Perform / Arp / Effects section.

An additional Edit mode is available to set preset attributes, comments and user tags so that you may customise the results list for specific projects or a live performance.

2.2 Simple Mode



The Simple mode or Performance page in Alchemy is divided into two sections, shown above.

1. The Title bar.
2. The Performance section.

2.3 Advanced Mode

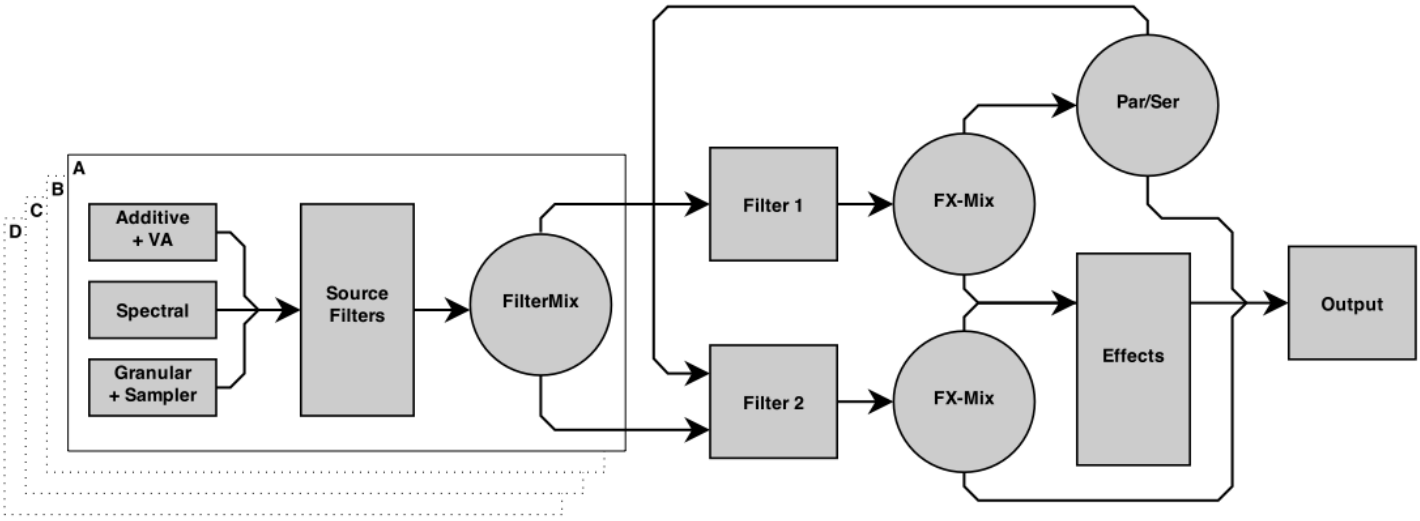
The main or 'global' page in Alchemy is divided into six sections, shown above.

1. The Title bar.
2. The Source section.
3. The Filter section.
4. The Master section.
5. The Modulation section.
6. The Perform/Arp/Effects section.

2.4 Signal flow in Alchemy

Before you can design new sounds with Alchemy you'll need to understand how its different parts fit together, and how each of them in turn contributes to what you hear.

At first glance Alchemy may seem complicated, but its layout is relatively simple:



This diagram represents signal flow in Alchemy, reading from left to right, through the different sound-generating and -processing modules.

These are the three basic stages:

1. There are four Sources (A, B, C, and D), each consisting of Additive, Spectral, Granular and Virtual Analogue elements, and a bank of Source Filters. These sources are used to create and shape the basic tone of the sound. All modulation in this section is polyphonic i.e. modulation is applied *per-voice*.

2. There are two Main Filters (1 and 2), which can be operated in parallel or in series. The main filters are used to shape or otherwise alter the combined sound from the four sources. All modulation in this section is polyphonic.

3. After the individual voices are mixed together and filtered, they pass through the Effects stage. Any modulation applied to the effects section is global to all voices i.e. modulation is no longer per-voice, but applied to the whole audio signal sent from the main filter section.

3 Using the Interface

While each group of functions in Alchemy is organised into a control panel or sub-panel described elsewhere in this Manual - see the table of contents at the beginning of this document - several types of controls are found throughout the interface. These include knobs, buttons, and value fields.

Alchemy also provides more specialised types of controls, such as the MSEG breakpoint editors and the graphical Source Editors. Each of these is explained separately on its own page in the Manual (MSEG, Source Edit, and so on).

3.1 Working with knobs

Many parameters are controlled via knobs. When you adjust a knob, its value is displayed and updated in the large Parameter Value display towards the top of the interface. Several useful techniques let you work with greater speed and accuracy.

- To **inspect a value** (without changing it), simply **hover over a knob** (without clicking it).
- For **fine control** of a parameter, **shift-drag** its knob.
- For **coarse but rapid control** of a parameter, hover over its knob and roll your scroll wheel or swipe the trackpad.
- To **reset any knob** immediately to its 12 o'clock position, **double-click** it.
- By default, all knobs in Alchemy respond to **linear dragging**: drag a knob upwards to increase it and downwards to decrease it.
    - You can **alt-drag** (PC) **or option-drag** (Mac) a knob to have it respond momentarily to **circular dragging**: drag a circular path around the knob, clockwise to increase it and counter-clockwise to decrease it. One nice application of this technique: you can set a knob **instantly** to the position you want by **alt- or option-clicking** that position.
    - If you prefer circular dragging as the default behaviour you can set a parameter in the configuration file. See the configuration section for details.
- To access a **contextual menu** offering various knob actions, **right-click** or **control-click** a knob.

3.2 Working with buttons and value fields

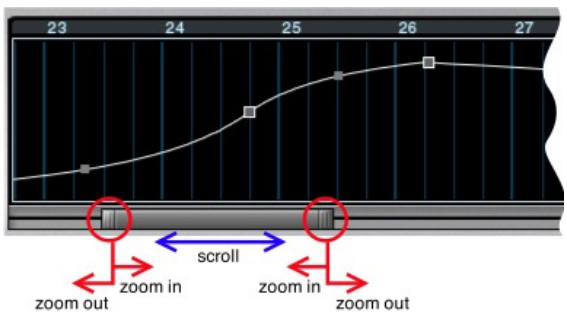
Operating buttons on the Alchemy interface is straightforward: simply click them. Many buttons behave as toggles: click a button once to turn a function on (button is lit) and click it again to turn the same function off (button is unlit). Some groups of buttons are mutually exclusive: turning one on turns another off. Some buttons contain pop-up menus: click one of these buttons and then choose a more specific command from the menu.

Value fields display their current value in a text field. Clicking in this field opens a **pop-up menu** from which you can choose a new value. Many value fields also provide **forward and back buttons**, so you can browse through the available choices.

Scrolling and zooming

Several of Alchemy's displays can be scrolled left and right to bring more controls or data into view, and many of these displays can also be zoomed in and out, so you can focus on small details or see more data at once.

- To **scroll**, click the scroll bar and drag it left or right or use the mouse wheel.
- To **zoom in**, click a zoom handle at either end of the scroll bar and drag it inward. To **zoom out**, drag it outward.
- To **zoom all the way out**, exposing a maximum amount of data, double-click the scroll bar.



3.3 MIDI learn

MIDI learn is used to assign MIDI Continuous Controller (CC) numbers to knobs on the GUI via an attached MIDI capable keyboard. One MIDI CC number is allowed per Alchemy control.

To activate MIDI learn for a control:

1. Right-click (control-click) any of Alchemy's knobs and choose 'MIDI Learn' from the contextual menu.

2. Move a control on your MIDI keyboard/controller.

Alchemy will automatically 'learn' the MIDI CC number and assign it to the knob. Now you can adjust the onscreen knob by operating the hardware control.

Every one of Alchemy's knobs can 'learn' a MIDI control in this way.

Alchemy's XY Pads and Remix Pad (see the Performance controls page) 'learn' in almost the same way, except that there are two options in the pop-up menu. 'MIDI Learn X' assigns a control to the pad's horizontal axis. 'MIDI Learn Y' assigns a control to the pad's vertical axis.

Once a MIDI control is learned, the assignment persists each time you load Alchemy. You can undo an assignment by right-clicking the knob and choosing 'MIDI Unlearn' from the contextual menu.

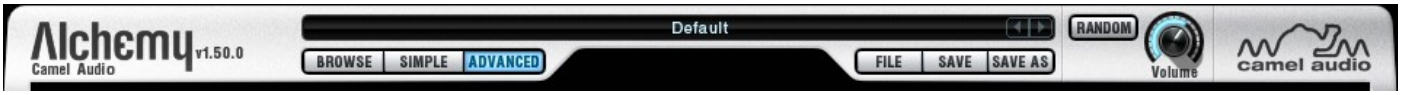
The current set of MIDI-learn assignments is stored in a file called 'Midimap.txt', which is located in the Alchemy data folder. The file is automatically updated, and is global to all instances of Alchemy.

Note: There is only one Midimap.txt file and it is global to Alchemy. Within a single host session you may use the same MIDI CC numbers to control different instances of Alchemy. However, these settings will not be preserved when you re-open the project.

The most recently learned CC number will be stored in the MIDI map file, overwriting any previous settings.

4 Title Bar

Alchemy's title bar provides tools for preset management and a few additional functions.



At the top and in the centre is the title bar, which is a single field displaying the current preset name and category, or 'default' if no preset is loaded.

Click the title bar to display a list of presets available to load, based on the current Preset Browser settings. By default, all presets are available to load. To quickly load a preset use the forward and back arrows adjacent to the Preset field to scroll through all available presets. The category of the selected preset will be displayed to the left of the preset name when a preset is loaded. The order of the list is determined by settings in the Preset Browser.

4.1 Parameter value display

Beneath the preset field is a large numerical display that shows the current value of the active or selected control, calibrated in appropriate units (e.g. '50%', '5 semis', '2859 Hz', and so on). Alchemy also uses this display for occasional messages (e.g. to confirm that a Save command was executed successfully) and to display the sound-designers name when a preset is first loaded.

4.2 Browse, Simple and Advanced buttons

When the **Browse** button is active the Preset Browser view is displayed. The **Simple** button displays a compact interface suitable for performing with Alchemy, consisting of the just Title Bar and the Performance section. When the **Advanced** button is active, a larger interface is displayed in which all sections of Alchemy are accessible; Advanced mode is best suited to sound design or detailed tweaking of existing presets. (See the Overview section for a quick tour of the entire interface.)

4.3 File, Save and Saveas buttons

The **FILE** button opens a pop-up menu containing several commands useful for preset management:

Load - opens a dialogue box where you can select a preset file (*.acp), SFZ file (*.sfz) or sample file (*.wav, *.aif) to load.

Save Consolidated - saves a copy of the active preset along with copies of any audio files it depends on, except for Alchemy Factory samples as these files are common to all Alchemy users.

Save CamelSounds – saves the current preset and all audio files as a zip file suitable for transferring to Alchemy Mobile or to install on another computer using the Alchemy Soundbank Installer app.

Clear - Initialises Alchemy, giving you a basic starting point for creating your own presets. (The initialised preset is a basic sawtooth wave.)

The **SAVE** button saves your changes to the currently active preset. If the preset belongs in the Factory bank the Save As dialogue box will be displayed so that you do not accidentally overwrite the preset.

The **SAVEAS** button opens a dialogue box where you can choose a file name (*.acp) and a location to save the currently active preset. When a new preset is saved it will be automatically added to the preset browser with the appropriate category, some attributes that are automatically determined and your sound designer name.

Notes about samples and preset data

- Audio samples will be referenced from the original folder unless Save Consolidated is used.
- Save Consolidated and Save CamelSounds are useful when sharing presets with other Alchemy users.
- Additive and/or spectral data will be stored as separate files (*.aaz).
- The first part of the .aaz filename is the same as the name of the saved preset, which ensures that the preset and the associated additive/spectral data appear side-by-side in an alphabetical directory.
- We recommend that you avoid renaming .aaz files in Finder/Explorer. Alchemy may fail to locate and load these files unless their original names are preserved.
- While renaming a whole preset in Finder/Explorer is less risky, this may lead to duplication of .aaz files to reflect the new name.
- In general, it's best to rename presets by using Alchemy's SAVEAS function. Once a preset is successfully saved under its new name, you can safely discard the old preset (*.acp) file.

4.4 Display options

Change Default GUI Mode - choose between Browser, Simple and Advanced modes as the default display when Alchemy is opened.

Skin – select the default skin to use for new instances of Alchemy. Changing the skin will prompt to close and reopen the GUI for changes to take effect. The setting is saved in the configuration file.

4.5 Other File menu options

About - display version number and licensing information.

Check for Update - check with the Camel Audio webserver to find out if you have the latest version of Alchemy installed.

Download Free Sounds - visit the Alchemy User Library on the Camel Audio website to download sounds created by other users.

Watch Tutorial Videos - visit the tutorials section of the Camel Audio website to view video tutorials illustrating how to use Alchemy.

Read the Wiki Manual - view the most up-to-date version of this manual online. The Alchemy interface also integrates with the online manual by right-clicking a control and selecting 'Help'.

Support Forum - visit the Camel Audio support forum on KVR Audio.

Newsletter - subscribe to the Camel Audio newsletter to receive update information and special offers.

Camel on Facebook - visit the Camel Audio Facebook page and Like us (because we rock!)

Buy More Sounds - visit the Camel Audio webshop to purchase new sounds for Alchemy.

Change Data Folder - set a new location for the Alchemy data folder on your hard disk.

Turn On iPad/iPhone Control - enable or disable remote control of Alchemy via Alchemy Mobile. The default setting is off.

Scan For New Presets - search the Alchemy data folder to update the preset browser. New sounds libraries and presets will be added to the results, presets that have been deleted will be removed.

MIDI Unlearn All – clears all MIDI Learn assignments and resets the Midimap.txt file.

4.6 Random button

Creates a random preset. Random preset generation in Alchemy is based on a collection of **preset templates**, arranged in the standard categories seen in the left hand browser column (Arpeggiated, Bass, Brass, and so on). If the preset browser category is set to 'None' Alchemy will choose a template from a random category, otherwise a template from the current category will be used. Alchemy then creates 'intelligent' random values for the parameters that are active in the chosen template.

If you want more control over the process, you can add or substitute your own preset templates; the details of this process are described below.

Tip: Be sure to save the random preset or host project if you like the sound, because once the random button is pressed again you won't get it back!

4.7 Volume knob

Boosts or cuts the overall preset volume. This adjustment is made at the very end of the signal path, so it controls the effects as well as the dry portion of the signal, and it acts on all voices at once. (Therefore, it's less common to modulate the Volume knob compared to the Amp knob on the Master page.)

4.8 Advanced: Creating your own randomisation templates

You can tailor Alchemy's random preset generation to your own needs by adding your own randomisation templates. Instructions for creating and using these files follow. *Details in italics are optional; they represent the guidelines followed by the factory randomisation templates.*

Start with an initialised preset (FILE > Clear).

Turn on as many **Sources** (A–D) as you want your new presets to use. *Factory templates typically use all four.*

In each source, turn on one or more **Elements** and choose an appropriate **synthesis type**; for instance, you might choose the Granular element and set it to SAMPLER mode. *Factory templates use combinations of VA, GRANULAR, and SAMPLER synthesis types; they avoid Additive synthesis and Spectral synthesis (RESYNTH, NOISE-RESYNTH) because these types may cause randomisation to proceed too slowly. Factory templates*

favour GRANULAR mode for Loops, Sound Effects, Soundscapes, Synth Leads, Synth Pads, Vocals, and Drums, and SAMPLER mode for other categories.

Set an appropriate **morph mode**. *Factory templates favour **morph xy** for Sound Effects and Soundscapes, and **xfade xy** for other categories.*

Turn on one or both of the **Main Filters** if you wish to use them; and turn on any **Source Filters** that you wish to use.

Choose a suitable trigger **Mode** (Always, Retrigger, Legato) and number of **Voices** for the category.

Set parameters in **AHDSR 1** to appropriate values for the category (e.g. a slow attack for pads).

Add up to five **effects** appropriate for the category and in an appropriate order. Alchemy turns these on/off at random, ensuring that at least one effect is on, when it generates new presets.

Make any **additional modulation assignments** that seem appropriate for the category. *Factory templates often assign LFO 1 to Master Fine Tune. For the Loops category, factory templates modulate the Position of each Source with a tempo synced Ramp Up LFO set to 8 beats. For the Organs and Vocals categories, factory templates modulate Master Amp with an LFO.*

Adjust **other knob settings** to values that seem typical for the category. Alchemy generates new presets by varying these settings around the values you supply.

Ensure that **no audio files** (WAV, AIFF, SFZ, AAZ) are loaded, and that **no performance controls** are assigned (FILE > Clear in the PERFORM section, if necessary). *The arpeggiator should be off unless you are creating an Arpeggiated template.*

Save the file in Alchemy/Libraries/PresetTemplates/*category* where *category* is an exact match to the name of an existing preset category (Arpeggiated, Bass, Brass, and so on). As long as your template is stored in the right location, you can name it anything you want. *Factory templates use generic names such as 'Default.acp' for the most basic template and 'Sample.acp' for a template based on the Sampler synthesis type.*

5 Performance Section

When **Simple** mode is active, or if the **PERFORM** button is illuminated in the Perform / Arp / Effects section, the Performance controls sub-page is displayed.

5.1 Perform Controls

The Performance controls sub-page provides eight knobs, two XY pads, and a set of 'ADSR' envelope knobs – a total of 16 controls. These can be assigned to modulate parameters in the same way as Alchemy's other modulation sources. In addition to these assignable controls, the Performance section has its own gain stage with a dedicated 'Snapshot Vol' control. All controls in the Performance section may be automated or MIDI learned. All performance controls are available as host automation targets.

Hint: the perform controls may be assigned to any GUI control including those which are not normally accessible as host automaton targets such as LFO and ADSR parameters.

The eight knobs can each be named, by clicking in the field below a knob and typing a name. Similarly, a pair of fields is provided for naming the X and Y axes of each XY pad. When a custom name is assigned, auto-assign will not update the name. Delete the perform control text to re-enable auto-assigned names.

Right-clicking (control-clicking) any perform knob opens a pop-up menu with the usual MIDI Learn/Unlearn commands, plus the following more specialised functions:

Delete Modulation - offers a list of all the targets receiving modulation from the knob; selecting an item in this list deletes the corresponding modulation assignment.

Swap With - allows you to exchange all the modulation assignments between the knob you've clicked and any other performance control selected from a list.

Copy Setting to all Snapshots - updates all eight snapshots so that the current value of the knob you've clicked replaces the stored value of that knob in each snapshot. For instance, if you want the set the master Sustain knob to 100% in every snapshot, simply turn the knob to the 100% position, right-click it, and choose this command to apply the new setting to all eight snapshots.

Auto-assign - sets a modulation target for the selected control and updates the label. See the Auto Assign section for more information.

Invert Knob - modify the range of the selected performance control and associated modulation targets so that the minimum and maximum values are swapped. Invert Knob is only available for Performance Controls 1-8.

Right-clicking (control-clicking) an XY pad opens a pop-up menu containing similar commands for both the X and the Y axis.

5.2 Remix pad

An Alchemy preset contains eight **snapshots** of Performance control settings, numbered 1 to 8, and accessible via the numbered grid at the far right of the Performance controls sub-page. By default, snapshot 1 is selected, and outlined with a white box.

You can **morph** between Performance snapshots in real-time by clicking and dragging the white box around the Remix Pad. The Performance control knobs will update smoothly, modulating the connected parameters as you go. **Try it**: play some notes while moving the Remix Pad.

Note: to select a snapshot precisely, double-click it. (See also Keyswitching, below.)

To make a new snapshot, set each of the Performance controls as desired, right-click (control-click) the remix pad, choose 'Store Snapshot', and select a number from 1 to 8. (Or choose 'All' to store the current Performance control settings in all eight snapshots. Note that this will **overwrite** all eight of the existing snapshots.)

Store Current Snapshot – stores the current Performance control settings into the current snapshot, as indicated by the bright remix pad square under the remix pad puck.

Store Snapshot - lets you store the current Performance control settings into the current snapshot, or into your choice of snapshots 1 to 8.

Swap Current Snapshot - lets you exchange the settings of the current snapshot with those of another snapshot of your choice.

5.3 Auto-assign and other controls

The **FILE** button on the Performance sub-page opens a pop-up menu with the following options:

Copy Snapshot - stores all the Performance settings of the currently selected snapshot in a buffer.

Paste Snapshot - pastes all the Performance settings from the buffer to the currently selected snapshot.

Copy 1 to All - copies all the Performance settings of snapshot 1 into snapshots 2 to 8.

Clear - captures the current values of the Performance controls and applies them directly to the target parameters, preserving the current sound; it then deletes all Performance control and Remix pad assignments, resets all the controls to zero, and clears the knob and pad name fields.

Randomise **Snapshots** - creates random performance control values for all snapshots except the currently active snapshot.

Auto Assign All - automatically assigns all the Performance controls to appropriate parameters in the current preset. It also intelligently creates a random set of snapshots, as variations on the settings in snapshot 1. This is the quickest way to get started with the Performance controls. Try it!

Auto Assign Empty - automatically assigns all the unused Performance controls to appropriate parameters in the current preset. It doesn't interfere with any Performance controls assignments you have already made, and it doesn't create snapshot settings.

Tip: By using the Remix pad **FILE** button commands it is possible to create entirely new variations from existing presets.

Load a preset.

Find a pleasing sound using the Remix pad.

Select Remix pad File -> Clear.

The 'found sound' from step 2 is retained, with all performance controls and Remix pad variations reset.

Select Remix pad **File** -> **Auto-Assign All**

A new set of Remix pad variations based on the found sound is now ready to play

5.4 Remix pad keyswitching

It's possible to switch between Performance snapshots using your MIDI controller keyboard.

From the Octave pop-up menu, select a MIDI note name (e.g. 'C4'). The first eight notes of that octave (including both white and black keys, e.g. C4–G4) will be assigned to snapshots 1–8. Hit one of the assigned keys on your controller to select the corresponding snapshot.

Note that MIDI notes assigned to snapshots in this fashion will not trigger normal notes in Alchemy. To ensure that notes play normally across the entire range of MIDI note numbers, choose 'None' from the Octave menu.

5.5 Modwheel

The Modwheel pop-up menu links the modwheel to one Performance control of your choice (any knob, or the X or Y axis of either pad).

Each modwheel target offers an inverted option. This is useful when using a spring loaded modwheel so that the target performance control does not close then the modwheel is released.

Tip: Assigning the modwheel to modulate any target in Alchemy is a two step process:

1. Assign one of the Performance controls to modulate the target.

2. Link the modwheel to the Performance control you assigned in step 1, by selecting that control in the Modwheel pop-up menu.

5.6 Recommended perform control assignments

The performance controls can, of course, be assigned any way you find useful. But if you are a sound designer creating presets for others to use, it is recommended that you follow the same guidelines that were established for Alchemy's factory presets. These guidelines, along with the rest of the Camel Audio 'House Style' for preset designers, are described on the Sound Designer Guidelines page.

6 Preset Browser

The preset browser is used for loading presets, searching, sorting, rating, and tagging presets.

6.1 Loading presets

When a new instance of Alchemy is created the browser results list will display all available presets. Clicking on a preset in the preset browser results list or the title bar preset list will load and highlight the preset. The title bar will update to display the preset category and preset name.

There are two different views for the preset browser results list: click the **Browser** button to open the Browser view, or click the preset name in the title bar to display a drop down list of presets. Both lists are synchronised, so if the preset browser is used to select a particular category or sort the results the title bar list will reflect your choices.

6.2 The browser results list

The browser results list displays all presets matching the current search criteria. The first column **Ratings** displays user ratings. The second column **Preset** displays the preset names. The number of presets returned by the current search criteria is displayed to the right of the Preset column header. In the example to the left a total of 3998 presets is listed.

6.2.1 Sorting the results list

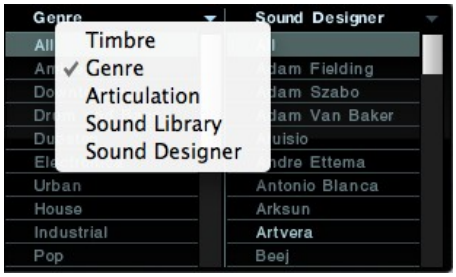
Clicking on the **Preset** or **Rating** column headers will sort the results list. A second click reverses the current sort order. In the example to the right, the Rating column was clicked. The highest rated presets appear first in the list. User rated presets always take precedence over unrated presets. Three dim stars indicate that a preset has not yet been rated by the user.

6.3 Attribute columns

Attributes are used to refine the preset results list and apply attributes to presets in Edit mode. For example, in search mode clicking **Guitars** in the **Category** column will limit the preset results list to include all presets belonging to the 'Guitars' category. Selecting an entry in the Category column will limit the choices in the Subcategory column to those applicable to the current Category.

It is possible to select multiple attributes from a single column using standard OS modifier keys. For example clicking **Guitars** and Cmd/Ctrl-clicking **Leads** will select both 'Guitars' and 'Leads' and include presets from both categories in the results list. Left-clicking category **Bass** and shift-clicking **Drums** would select categories 'Bass', 'Brass' and 'Drums'.

6.3.1 Additional attribute columns



There are six attribute columns available. In addition to the Category and Subcategory columns that are permanently visible, the attributes displayed in the four right-hand columns may be changed by clicking the column headers. A menu is displayed, offering a choice of: Timbre, Genre, Articulation, Sound Library and Sound Designer.

Each column includes an 'All' value at the top of the list. Click 'All' to to remove the selected attribute(s) from the selection criteria for that column. In the above example clicking 'All' in the Sub-category column would remove **Bowed** and **Electric** from the search criteria, increasing the number of **Guitar** presets displayed in the results lists.

6.4 Rating presets

To **rate** a preset the preset must be selected in the preset browser results list. Click a star in the **Ratings** column to set the rating between 1 and 5. The rating is immediately applied to the preset. **Unrated** presets are identified by three blue stars.

It is possible to limit the preset results list to match any star rating by entering the number of stars in the text search field. Searching for '****' will display only 4-star rated presets.

6.5 Preset properties

When a preset is loaded the area to the right of the preset browser results list displays an image for the preset bank, the sound designer's name and any user tags or comments that have been applied to the preset.

The **Edit** button is used to enter Edit mode.

6.6 Text searching

The text field above the sound designer's name is used for **manual searching** of presets by name, attributes, user comments or tags. Enter your search term and click the **magnifying glass** (or press **Enter**) to filter the preset browser results list. In the above example the search term 'space project' is used. The first preset in the results list is loaded, and we can see the 'User Tags' field matches the search term.

6.6.1 Wild card searches

Text searches always add a **wild-card** character to the end of each search term. Searching for 'Pad' will match 'Pad' and 'Pads' but not 'Pa', 'Par' or 'Parrot'.

The search term is always saved and recalled with the host project, making it well suited for creating custom MIDI program change lists or project templates with a predefined set of presets available.

To clear the current search term click in the search field, ensure all of the text is selected, and press the delete key.

6.6.2 Negated search terms

Search terms may be **negated** using the '-' symbol. For example, to search for all presets matching 'analog' but

not 'bass' use the search term 'analog -bass'. This may also be used to remove all 'bass' presets from the results list by entering only '-bass' in the search field. Presets in the Bass category and presets with 'bass' or 'basses' in the preset name, User Tags or Comments field will be removed from the preset results list. Negated search terms may be combined with attributes to further refine the results list.

6.7 Selecting presets using MIDI Bank and Program Change messages

MIDI Program Change messages select a preset in the current preset browser results list. The index of the program change message is used to determine which preset is selected, starting from 0. For example program change number '0' will select the first preset in the list. Program change number '2' will select the third preset (remember, we are counting from 0). The maximum program change index is 127, allowing for a total of 128 presets to be selected within the current bank.

Bank messages are used to select presets when the number of presets in the preset browser results list is more than 128. A bank change message must be followed immediately by a program change message. The bank number is multiplied by 128; the program change value is then added to the resulting bank value. Some examples:

Bank number 0 and Program Number 9 will select preset number 10 (0 x 128 + 9).

Bank number 2 and program number 9 will select preset number 265 (2 x 128 + 9).

The **current bank** will remain active until a new bank and program change pair is received, so the next program change message received will be added to 256 using our example of Bank number 2. When the bank number or program number exceeds the total number of presets in the current results list the message will be ignored.

All browser attribute selections and the search term are recalled with your host project so that MIDI Bank and Program Change messages will reliably select the correct preset during playback or a live performance. However, installing or saving new presets may cause the results list to change the next time the project is recalled. For accurate results list recall we recommend tagging presets required fora specific performance to avoid any nasty surprises!

6.8 Edit mode

Clicking the **Edit** button places the preset browser in preset edit mode. All editable attribute columns are displayed and the preset browser extends to cover the perform section, making attribute selection easier. It is possible to select multiple presets using standard OS modifier keys e.g. Shift-click will select a range of presets. Selecting multiple presets for editing allows you to quickly apply attributes, User Tags or Comments to groups of presets.

All changes made while Edit mode is active are applied immediately.

Click the **Edit** button again to exit Edit mode. The preset browser will return the main view.

6.8.1 Applying New Attributes, User Tags and Comments

While **Edit** mode is active any attributes selected in the browser attribute columns, or text entered in the **User tags** or **Comments** fields, will be applied to the preset and automatically saved to the preset database.

Changing the preset category will only update the preset category; it will not move the preset to the new category folder on your disk. It is not possible to change the Sound Library for a preset, this column is displayed for reference only.

Click in the **User Tags** or **Comments** field to enable text entry. Any terms entered in these fields may be searched using the text search facility. This is useful for tagging presets for a particular project, or to create custom MIDI program change lists for a live performance.

While **Edit** mode is active the Previous and Next preset selectors in the Title Bar may be used to change presets. All attributes, the User Tags and Comments assigned to the preset will load along with the preset. The Subcategory column will update to display only the sub-categories available in to the current Category.

6.9 Backup or transfer preset ratings, comments and tags

Ratings, comments, tags and attributes are stored in the preset database file. The file is found in the Alchemy configuration folder (see the manual conventions near the start of this document for details of where this folder exists on your system). The following files which should be copied to a safe place at regular intervals:

```
Alchemy_Preset_Ratings_And_Tags
Alchemy_Preset_Ratings_And_Tags.bak
```

Should you need to re-install Alchemy (for example, on a new system) and want to retain your preset ratings and edits the above files should be restored to the configuration folder before starting Alchemy for the first time. A preset scan will initiate automatically, or select 'Scan For New Presets' from the File menu. The browser will be populated with the presets from the restored DB file.

7 The Arpeggiator

Alchemy offers a flexible and powerful **Arpeggiator**. To access it, click the **ARP** button in the Perform/Arp/Effects section towards the bottom of the interface. A large number of controls makes the Arpeggiator unusually versatile; but if you just want to create classic arpeggiator patterns, you'll find this very easy to do. (In fact, you can probably skip straight to the Examples at the bottom of this page!)

The Arpeggiator sub-page consists of three main sets of controls. On the left is a series of nine knobs that control the basic behaviour of the Arpeggiator. In the middle is a set of buttons, pop-up menus, and knobs that let you manage one or more Sequencers. These are called 'Arp Sequencer' modules, because they are internal to the Arpeggiator and separate from Alchemy's normal Sequencers. On the right of the Arpeggiator subpage is a step editor for the Arp Sequencer modules. Below, you will find details about each of the basic sequencer controls, and an explanation of what the Arp Sequencer modules do. For more details about individual Arp Sequencer controls, see the explanation of the same controls on the page describing Alchemy's normal sequencers.

7.1 Basic controls

The basic **Arpeggiator** controls consist of nine knobs. Here's what each of them does.

Mode. Turns the Arpeggiator On and Off, and determines the order in which incoming notes are organised into a pattern. In addition to 'Off', you have the following choices:

Up - plays the current notes from lowest to highest.

Down - plays the current notes from highest to lowest.

Up/Down - lowest to highest and back again.

Down/Up - highest to lowest and back again.

As Played - plays the current notes in the order they were originally played.

Random - plays the current notes in a random, non-repeating order.

Chord - plays all of the current notes simultaneously as a chord. In this mode, the chords you play will 'pulse' according to the Arpeggiator's rate and rhythm, as determined by the Arp Sequencer settings.

> **Note**: Like most knobs throughout the Alchemy interface, Mode is a mod target. In several of the factory presets, a Perform knob is set up to modulate Mode, so you can easily turn the Arpeggiator on and off during a performance.

Latch - When Latch is Off, the current Arpeggiator pattern will stop playing when you lift your hands off the keyboard. When Latch set to **Hold** or **Add**, the Arpeggiator 'holds' notes for you, so the current pattern will continue to play when you lift your hands off the keyboard. If you then play one or more new notes, two results are possible:

In **Hold** mode, the new notes are organised into a new pattern, which replaces the existing pattern.

In **Add** mode, the new notes are added to the existing pattern. (Try combining 'As Played' mode with the 'Add' latch setting to create an interactive step sequencer with up to 128 steps!)

Amp. - The Amp knob determines the velocity of notes played by the Arpeggiator (but see also the KeyVel knob). It can be modulated to create variations in note velocity as the Arpeggiator plays.

> **Note** that Arp Sequencer 1 is assigned by default to modulate the Amp knob. Normally you should not remove this modulation assignment. Even if you don't need the step values of Arp Sequencer 1 to determine the velocities of notes played by the Arpeggiator, this modulator plays a more fundamental role: the rising edge of each Arp Sequencer step is the Arpeggiator's cue to play the next note in its current pattern.

Tune - The Tune knob applies a pitch offset to all of the notes played by the Arpeggiator. For example, if you set Tune to 12 semitones, the Arpeggiator will play patterns an octave higher than the MIDI input. Tune is also available as a target to be modulated (e.g. by an Arp Sequencer). This allows you to create a variable pitch offset as the Arpeggiator plays.

Pan - Sets a position in the stereo field for all the notes played by the Arpeggiator. It can be modulated to create variable panning as the Arpeggiator plays.

Split - The keyboard can be split so that notes below a certain point are fed to the Arpeggiator, while notes above that point play normally. Split determines the highest note to be included in the Arpeggiator pattern; set it to the maximum value 'g8' to arpeggiate all of the MIDI input.

KeyVel - Arp Sequencer 1 modulates the Arpeggiator's Amp knob by default, and when KeyVel is 0%, the step values of Arp Sequencer 1 fully determine the velocities of notes in the Arpeggiator pattern. When KeyVel is 100%, these velocities are determined instead by the incoming MIDI data, so if you strike a key hard the corresponding note in the Arpeggiator pattern will have a high velocity level. You can also set KeyVel to intermediate values to blend the Arp Sequencer step values with the velocities in the incoming MIDI data.

Octave - Determines whether the Arpeggiator pattern is played only at its original pitch level, or is repeated across additional higher octaves.

Source - When Source is set to 'All', the Arpeggiator pattern is played by all sources (assuming they are all turned ON). You can also restrict the Arpeggiator pattern to any one source by setting this knob to a value of 1-4; then the other three sources will play the incoming MIDI data normally.

> **Note**: When Source is set to a value other than 'All', so that one voice is arpeggiated while the others play normally, Alchemy uses separate voices to play the arpeggiated and non-arpeggiated portions of its output. Therefore, this feature requires the number of Voices to be set to at least 2 in the Master section.

Arp Sequencer modules work just like normal Sequencer modules in Alchemy. They have a more specialised job to do, however, because they are internal to the Arpeggiator and are designed to modulate its basic controls (which are described above).

> **Notice** that if you right-click a knob in most sections of Alchemy (e.g. a Cutoff knob in the Main Filter section) and peek at the 'Add Modulation' options in the contextual menu, your choices include 'Sequencer'. But if you right-click one of the Arpeggiator's basic controls (e.g. its Pan knob), your choices include 'Arp Sequencer' instead.

By default, the Arpeggiator is configured with just one Arp Sequencer, and for most purposes this is all you will need. Arp Sequencer 1 modulates the Arpeggiator's Amp parameter. This modulation assignment enables the Arpeggiator to 'watch' Arp Sequencer 1: whenever the Arpeggiator sees the rising edge of another Arp Sequencer step, it plays the next note of its own current pattern.

Various settings in the Arp Sequencer allow you to adjust the Arpeggiator's behaviour.

Duration - Choose a different 'Duration' to make the Arpeggiator play faster or slower.

Shuffle - Lets you create various 'swing' effects. Setting a Shuffle value greater than 0% increases the duration of the odd-numbered step (1, 3, 5, …) and decreases the length of the even-numbered steps correspondingly.

> **Note**: For classic 'swinging sixteenth notes', set Duration to 1/4 and try a Shuffle value of around 20% (light swing) to 30% (heavy swing). You can get more extreme effects using even higher Shuffle settings; at 100%, the even-numbered steps get so short that they disappear entirely!

Sustain - sets the length of each note played by the Arpeggiator, as a percentage of the duration. Lower values of Sustain will give you more 'staccato' results. (You may not hear a staccato effect, however, if your preset has a long release time.)

The Arpeggiator's basic controls and the Arp Sequencer settings described above add up to a well-rounded Arpeggiator - one that can create a variety of patterns using notes played on the keyboard or other MIDI input, with controls in the Arp Sequencer for speed (Duration) and rhythmic feel (Shuffle, Sustain). If you want even more options, you'll find them in the Arp Sequencer step editor.

7.2 Using the Arp Sequencer step editors

When an Arp Sequencer modulates the Arpeggiator Amp, its step editor can be put to a number of uses. (Recall that Arp Sequencer 1 modulates Amp as part of the default configuration of the Arpeggiator.)

You can edit the Arp Sequencer step **values** to create a pattern that the Arpeggiator will apply to the velocities of the notes it plays.

If you set the value of a step to 0%, the Arpeggiator **won't play a note** at that step. (There's no 'rising edge' for the Arpeggiator to see when the step value is 0%; therefore it doesn't trigger a note.) This creates a gap in the rhythmic pattern - a 'rest', in musical terms.

Similarly, you can 'tie' one step to the next by shift-clicking below it. A small 'chain-link' symbol appears below each tied step. When two or more steps are tied together they behave like one longer step; only the first step triggers a note in the Arpeggiator, while subsequent tied steps provide sustain.

You can edit the Arp Sequencer step **lengths** to create a pattern of shorter and longer note lengths. These values are combined with the overall note length determined by the Arp Sequencer **Sustain** control.

You can edit the Arp Sequencer **swing** at each step to create subtle (or not-so-subtle) variations in timing. Each swing value ranges from 0 to 2; the middle value of 1 represents normal timing, while smaller values play earlier and larger values play later than normal. These swing values are combined with the overall timing pattern determined by the Arp Sequencer **Shuffle** control.

You can apply additional sets of Arp Sequencer steps to other basic controls in the Arpeggiator, such as Tune and Pan. Simply right-click on one of these knobs and choose 'Add Modulation' > 'Arp Sequencer' > 'New Arp Sequencer'. (Indeed, that's an early step in the example entitled 'Creating a step sequencer', found at the bottom of this page.)

> **Note**: 'New Arp Sequencer' creates a new set of Value, Length, and Swing patterns that you can edit to suit your wishes; but all of the Arp Sequencer modules share a single set of controls such as Duration, Shuffle, Sustain, Trigger status, and NumSt (number of steps). Alchemy's normal Sequencer modules work differently: each one has an independent set of controls such as Duration.

7.3 Importing from a MIDI file

Alchemy's MIDI-import capabilities are described on the Sequencer page. The same **Import Velocity**, **Import Note**, and **Import Groove** commands are available here in the Arp Sequencer modules. An additional command, **Import All** is also provided; it sets step values in Arp Sequencer 1 based on extracted velocity data, sets swing values in Arp Sequencer 1 based on extracted groove data, sets step levels in Arp Sequencer 2 based on extracted note data, routes Arp Sequencer 1 to the Arpeggiator's Amp knob with a depth of 100%, and routes Arp Sequencer 2 to the Arpeggiator's Tune knob with a depth of 24 semis. The resulting configuration allows the Arpeggiator to play back the imported MIDI notes, velocities, and groove timing faithfully.

7.4 Arpeggiator examples

7.4.1 Creating classic arpeggiator patterns

Initialise Alchemy by clicking **FILE** in the Title Bar and choosing 'Clear' from the pop-up menu.

Switch the Perform/Arp/Effects section to **Arp** at the bottom of the interface, and play and hold a chord. Set **Mode** to 'Up/Down' and **Latch** to 'Hold'. Now when you play a chord, it will be processed by the Arpeggiator, which continues to play when you release the chord.

Next, as the Arpeggiator continues to play, you can explore additional Arpeggiator parameters:

Try changing the **Mode** to a value other than 'Up/Down' - 'Up', 'Down', 'Down/Up', and' 'Chord' are among the most popular additional Arpeggiator modes.

Try setting **Octave** to a value greater than one in order to have the Arpeggiator pattern play in more than one octave range.

Try changing the Arp Sequencer **Duration** from its default value of '1/4' to '1/2' (so a note plays every half beat), to '1/8' (eight notes per beat), and to other values. Restore a value of '1/4' or greater before trying the next step.

Experiment with the Arp Sequencer **Shuffle** and **Sustain** parameters.

While the Arpeggiator continues to run, you can also adjust settings in other sections of the interface. For instance, try the following adjustments:

Source A **VA waveform** = 'Basic' > 'Square'; **NOsc** = 2.

AHDSR 1: ***Decay*** = 0.50 sec; ***Sustain*** = 0%; ***Release*** 1.5 sec.

Main Filter 1: ***Cutoff*** = 550 Hz; ***modulate*** Cutoff with AHDSR 1 (depth = 30%).

7.4.2 Example: Creating a step sequencer

Initialise Alchemy by clicking **FILE** in the Title Bar and choosing 'Clear' from the pop-up menu.

Switch the Perform/Arp/Effects section to **Arp** at the bottom of the interface, and set **Mode** to 'Up' and Latch to 'Hold' Now play a single note and the Arpeggiator will play that same note repeatedly as you work through the rest of this example. (If this drives you crazy, set Latch to 'Off' instead!)

Right-click on the Arpeggiator's **Tune** knob and choose 'Add Modulation' > 'Arp Sequencer' > 'New Arp Sequencer'.

The Tune knob's mod rack appears in Alchemy's MOD section, and 'Arp Sequencer 2' can be seen in the first slot of the rack.

Adjust the modulation Depth knob, to the right of this slot, to a value of 24 semitones in order to program a step pattern spanning two octaves.

Returning to the Arpeggiator sub-page, set the 'Snap' value for the Arp Sequencer step editor to 1/24 in order to snap step values to semitones (since the modulation depth is 24 semitones).

Ensure that the step editor view is set to Value (rather than Length or Swing, which are only useful when you're modulating Amp). Now you can create a pattern of pitches by editing each step value as you desire.

7.4.3 Make it stop!!!

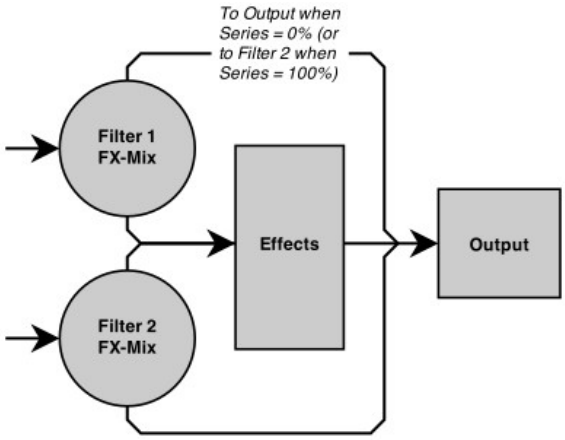
If the Arpeggiator keeps playing and you want it to stop, do one of the following things:

Set the Arpeggiator **Mode** knob to 'OFF'.
Set the Arpeggiator **Latch** knob to 'OFF'.
Stop (or Play and then Stop) your host's transport.

8 Effects Rack

Alchemy's Effects module provides a powerful multi-effects processor, offering up to five high-quality effects simultaneously. The available effects include all of the choices found in Camel Audio's popular CamelPhat and CamelSpace multi-effects plug-ins as well as its Cameleon 5000 synthesiser. Several new effects, including a highly realistic Acoustic Reverb, round out the choices.

The Effects module receives its input from the two Main filter modules, in varying amounts, according to their respective FX-Mix settings.



It sends its output directly to Alchemy's main output.

> **Note**: the Effects module's output is mixed together with the 'dry' output from the Main filter modules before arriving at Alchemy's main output.

8.1 Global page controls

When the **EFFECTS** button is illuminated, the Effects sub-page is displayed in the Perform / Arp / Effects section.

The left-hand side of the Effects sub-page contains the **effects rack**, with five slots into which effects can be loaded. The signal flows through these loaded effects **from the top to the bottom** of the rack, so you can get different results by loading the same effects in different orders.

Clicking the **FILE** button above the effects rack gives you access to a pop-up menu with several useful commands:

The **Load, Save**, **Copy**, and **Paste** commands act on the entire contents of the effects rack and on the settings of each loaded effect.

The **Clear** command removes all the loaded effect effects, giving you an empty effects rack.

The **Randomise** command randomises the settings of all the currently loaded effects.

Each slot of the effects rack consists of an **on/off button** and a **selection field**.

To **bypass an individual effect**, click its on/off button (lit = on, unlit = off).

To **remove an effect** from the rack, click its selection field and choose 'None' from the pop-up menu.

To **load an effect** in the rack, replacing any previously loaded effect in the same slot, click the selection field and choose from the pop-up menu.

Each effect can be used in only one slot, but a second instance of several effects is provided. (For instance, if you've already used the Delay1 effect, it will be greyed out in the pop-up list of effects available in the other effects rack slots; but you can still choose Delay2 further down in the list.)

To **change the order of the loaded effects**, right-click (control-click) a selection field and use the 'Move Up' and 'Move Down' commands.

8.2 Controlling individual effects

For each effect loaded in the effects rack, there is a corresponding **control panel** in the scrolling window on the right-hand side of the Effects sub-page. (The top-to-bottom order of the rack is matched by the left-to-right order of the window.)

Many of the effects control panels include **FILE** buttons. Clicking one of these buttons gives you access to a pop-up

menu with **Load**, **Save**, **Copy**, **Paste**, and **Clear** commands, which apply to the settings of the individual effect. (Note that the 'Clear' command restores an effect's default settings; it doesn't bypass the effect or remove it from the effects rack.)

Acoustic Reverb. High quality, full-featured reverb effect.

- **Time** - controls the length of the reverb tail; values up to 20 sec are possible.
- **PreDelay** - sets amount of initial delay before diffuse reflections begin.
- **Size** - determines the dimensions of the simulated space.
- **Width** - determines how much reflections are spread across the stereo field.
- **Diffusion** - determines the density of the early reflections.
- **Gate** - sets a threshold below which the reverb tail will be gated. Attack shapes the beginning of the gated tail, while Decay shapes its end.
- A built-in **EQ** lets you modify the frequency content of the wet signal. **LoFreq** sets the frequency of a low shelf that you can boost or cut with **LoGain**. **HiFreq** and **HiGain**, similarly, work like a high shelving EQ.
- There are four controls for **damping** (frequency-specific losses). **DampLoF** sets the frequency of a low shelf that you can damp by increasing **DampLoAmt**. **DampHiF** and **DampHiAmt**, similarly, provide high-shelf damping.
- **Variation** - provides subtly different 'colours' of reverberation.
- **Quality** - at lower settings of Quality, the Acoustic Reverb uses less CPU. At higher settings, it uses more CPU in order to provide a denser, smoother tail. When you need to conserve CPU, don't be afraid to try lower settings of Quality; for many types of material these settings still sound excellent.
- **Mix** - determines the wet/dry balance (0% = dry only; 50% = equal mix; 100% = wet only).

Camel Reverb. Reverb effect featuring the CPU-efficient algorithm used in Camel Audio's popular CamelSpace effect and Cameleon 5000 synthesiser.

- **PreDelay** - sets amount of initial delay before diffuse reflections begin.
- **Damping** - higher values mean more high-frequency loss in the reflections.
- **HighCut** and **LowCut** - set the frequencies above and below which the wet signal is cut.
- Take care not to inadvertently cut the entire wet signal by setting **LowCut** above **HighCut**.
- **Size** - determines the dimensions of the simulated space; larger values mean longer reverb times.
- **Mix** - determines the wet/dry balance (0% = dry only; 50% = equal mix; 100% = wet only).

Delay1 / **2**. Stereo delay with dual-filtered feedback.

- **L/R Rate** - sets the delay time, ***in msec*** when the **SYNC** button is off or ***in beats*** when the **SYNC** button is on.
- **L/R Offset** - adds a small additional amount of delay, so you can adjust the 'feel' of tempo-synced delays.
- **L/R Feedback** - determines how much of the delayed signal is fed back to the input of the delay.
- **Filter A, Filter B** - filters the delayed signal without affecting the dry signal. Toggle one or both filters with the **A** and **B** buttons (button illuminates when filter is on). Select a ***type*** for each filter via the selection fields. Each filter has **Cutoff** and **Resonance** controls.
- **SYNC** button - synchronises delay rates with the host tempo (See **L/R Rate**, above).

- **MONO** button - mixes the left and right input channels down to mono and feeds the result to both the left and the right channels of the delay. The 'dry' portion of the signal remains in stereo.
- **Crossover** - determines stereo placement of the feedback signal. (At 0%, left feeds left and right feeds right; at 50%, each channel is fed to both inputs; at 100%, left feeds right and right feeds left.)
- **Initial Pan** - determines the stereo placement of the initial delayed signal (prior to Feedback). Typically, you would set Initial Pan to 0% or 100% when Crossover = 100%, and leave Initial Pan centred 50% otherwise.
- **Mix** - determines the Wet/Dry balance (0% = dry only; 50% = equal mix; 100% = wet only).

Tip: the delay Filters A and B are configured in series, and the left and right channels pass through both filters.

Also: please be aware that some filter types and settings - such as medium-to-high Resonance – will boost the delayed signal. This can result in an **endless delay** feedback loop and very high frequencies. Lower the resonance or reduce the amount of Feedback in order to compensate for the boost.

Note: see the examples at the end of this section for an explanation of how to use Crossover and Initial Pan for a classic 'ping-pong' delay effect.

Mod FX1 / 2. Short delay with built-in LFO modulation, useful for **chorus**, **flanging**, and related effects.

- **Delay** - sets the base delay time. The shortest values are useful for flanging, values in the range of 10 to 40 msec are useful for chorus, and longer delay times can produce a variety of metallic and buzzing effects.
- A built in LFO drives modulation of the base delay time, and controls are provided for the LFO **Rate** and the **Depth** of modulation. Faster Rates and smaller Depths are characteristic of chorus effects, while slower Rates and greater Depths are typical for flanging.
- **Feedback** - mixes the delayed signal back into the input. Medium to medium-high settings are common for flanging effects, while chorus tends to use little or no feedback.
- **Stereo** - spreads the delayed signal across the stereo field.
- **Mix** - determines the wet/dry balance (0% = dry only; 50% = equal mix; 100% = wet only).

Distortion1 / **2**. Distortion effect with multiple algorithms that can be used simultaneously.

- **Tube** - simulates the warm distortion effect of an overdriven tube amp.
- **Mech** - produces a 'nastier', more intense flavour of distortion
- **Bit Crush** - a lo-fi digital flavour of distortion.
- **Xcite** - refreshes the high-frequency range.
- **Post Gain** - some distortion types, especially Tube and Mech, can boost the signal significantly; this control allows you to trim the output level to compensate.

Three Band EQ1 / 2. Offers three bands of parametric EQ. Each band has an identical set of controls.

- **LoGain** (**MidGain**, **HiGain**) - sets the amount of boost or cut to the Lo (Mid, Hi) band.
- **LoFreq** (**MidFreq**, **HiFreq**) - sets the centre frequency of the Lo (Mid, Hi) band.
- Note that LoFreq, MidFreq, and HiFreq can each be set to values from 16 Hz through 16744 Hz, so it isn't necessary (although it may be useful) to tune Mid above Lo, or Hi above Mid.
- **LoBW** (**MidBW**, **HiBW**) - sets the bandwidth of the Lo (Mid, Hi) band.

Bandpass Filter1 / 2. Adjustable-width bandpass filter; the rejected portion of the signal can be recovered at a later point in the signal path using the Band Reject (or Band Reject 2) module.

- **Low**, **High** - set the lower and upper edges of the passband, respectively.
- **LowRes**, **HighRes** - add resonant emphasis to the passband edges.
- **Band Reject1** / **2**. Only available when the Bandpass Filter1 / 2 is inserted upstream in the signal path. Allows you to mix in the portion of the signal that was rejected by the bandpass filter. A classic application is **multi-band distortion**; see the example at the bottom of this page.

- **Mix** - determines the how much of the band-rejected signal is mixed back into the signal path, bypassing any effects that sit in between the Bandpass Filter and the Band Reject mix point.

MM Filter1 / 2. Multi-mode filter, offering all the same filter types as Alchemy's Main Filters. Note that each voice in Alchemy is processed by its own Main Filters, while all the voices are mixed together before passing through the MM Filter and other effects.

- The **filter types** and the **Cutoff**, **Resonance**, and **Drive** controls are the same as for Alchemy's Main Filters.
- **Mix** - determines the wet/dry balance (0% = dry only; 50% = equal mix; 100% = wet only). Typically this is left at 100%.

Bass Enhancer. Easy-to-use, bass-boosting EQ, recommended for use with bass and low-frequency percussion sounds.

- **Amount** - controls the intensity of the bass enhancement.
- **Tune** - adjusts the frequency band to which the boost is applied.
- **P** (phat mode) button - adds a saturation effect, for a different 'flavour' bass enhancement. For more neutral/transparent results, leave phat mode off.

Compressor. Easy-to-use dynamics processor based on a soft-limiting compression algorithm.

- **Amount** - controls the amount of compression. Because make-up gain is built in, larger amounts of compression result in greater apparent 'loudness'.
- **Release** - controls how quickly the compression effect subsides once the input signal falls below the threshold for compression.
- **P** (phat mode) button - adds a saturation effect, for a different 'flavour' of compression. For more neutral/transparent compression, leave phat mode off.

Gain. Allows for a variety of auto-gate effects when modulated by an LFO or sequencer.

- **Gain** - a setting of 100% preserves the input signal level, while lower values reduce this level, all the way to silence at 0%.

Pan. Allows for a variety of auto-pan effects when modulated by an LFO or sequencer.

- **Pan** - a setting of 50% preserves the balance between the left and right input channels. Lower values boost the left while cutting the right; higher values boost the right while cutting the left.

8.3 Effect examples

8.3.1 Ping Pong Delay and Flanger

Initialise Alchemy by choosing the 'Clear' command in the Title Bar **FILE** menu.

Click the **EFFECTS** button to bring the Effects sub-page into view. Play a few notes at this point, just to get familiar with the '**dry' sound** of the default preset before any effects are applied.

Click in the first (top) slot of the effects rack and choose 'Delay'. Then play a few more notes and notice that the Delay module echoes what you play at a time interval of one beat (as determined by the host tempo). Now **bypass** the Delay by clicking the button at the left of the first slot in the effects rack.

Next, click in the second slot of the effects rack and choose 'ModFX'. Then play a few more notes and notice that the default settings of the ModFX unit produce a pleasant chorus sound. Let's adjust the ModFX settings to achieve a flanger effect. Try the following adjustments:

- Reduce the ModFX **Delay** time to approximately 0.0006 sec.
- Set the modulation **Rate** to around 0.2 Hz, and set the modulation **Depth** to 0.003 sec.
- Increase **Feedback** to 60% (or more, to taste).

Now that the flanger effect is working for us, use the buttons at the far left of the effects rack to bypass the ModFX module and re-activate the Delay so we can work on setting up a ping-pong delay effect. Try the following adjustments to settings in the Delay module in order to pan delays alternately left and right:

- Increase the left and right **Rate** settings from 1 beat to 1/2 beat.
- Increase **Crossover** to 100%.
- Reduce **InitialPan** to 0%.

Use the button at the far left of the effects rack to re-activate the ModFX module, so you can hear the flanger effect **combined** with the ping-pong delay. Notice that, on one hand, the ping-pong delay is most noticeable when you play short notes, but on the other hand, the sweep of the flanger is not very dramatic on these short notes. Let's reverse the order of the two effects to see if the situation improves.

To swap the order of the two effects-rack slots, right-click (control-click) the first slot and choose 'Move Down' from the contextual menu - or right-click (control-click) the second slot and choose 'Move Up'. In this new configuration, the sweep of the flanger can be heard across all the delayed copies of the sound, producing a more dramatic sweeping effect.

8.3.2 Multi-band Distortion using Bandpass Filter and Band Reject

Initialise Alchemy by choosing the 'Clear' command in the Title Bar **FILE** menu. Click in the Source A content field and choose 'Load Audio' > 'Factory' > 'Loops' > 'DrumLoops' > 'Silkloop'. Switch the Source A Granular element from Granular mode to Sampler mode for straightforward sample playback. (See 'Using Sampler mode' on the Granular page for details.)

The 'Silkloop' drum pattern should loop smoothly when you play and hold middle C. Now **our goal** is to 'beef up' the kick drum with Tube distortion, while keeping the higher-frequency elements 'clean'.

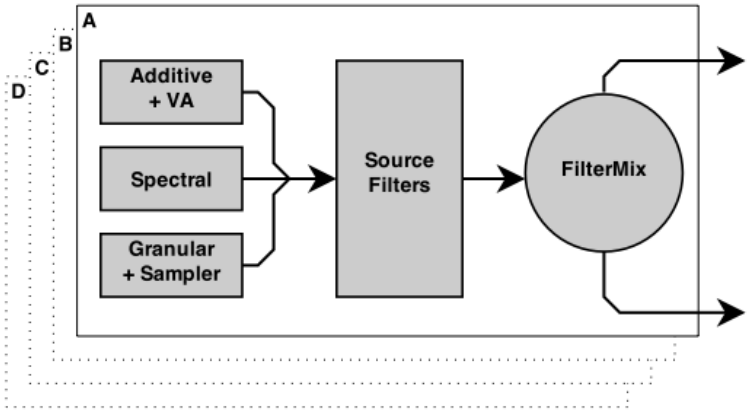
First, insert the **Bandpass Filter** into the first slot of the effects rack. While the drum pattern loops, reduce the High control until the kick drum is more-or-less isolated. (A value around 160 Hz works well.)

Next, insert the **Distortion** module into the second slot of the effects rack. Turn the Tube control all the way up to 100%. (You can add an extra 'edge' to the resulting distorted kick sound by increasing the Bandpass Filter's HighRes knob - try a value around 20%.)

Finally, it's time to bring the mid- and high-frequency portions of the drum back in via the **Band Reject** module. Simply insert this module in the third effects-rack slot, and turn the Mix control all the way up to 100%. As the drum loop continues to play, you can bypass and re-activate the Distortion module (using the button at the far left of its effects-rack slot) in order to confirm that this module is processing only the kick drum.

9 Sources

Alchemy's sound-generating modules are called **Sources**. There are four Sources available in an Alchemy preset (A, B, C, and D). Each Source is made up of an identical set of components:



The four Sources are entirely independent of one another, and can contribute to the overall sound in very different ways.

9.1 Source elements

Each individual Source provides **three sound-generating elements** based on different methods of synthesis:

An Additive element, which can also be operated in virtual analog (VA) mode.

A Spectral element, which can also be used as a raw white and filtered noise source.

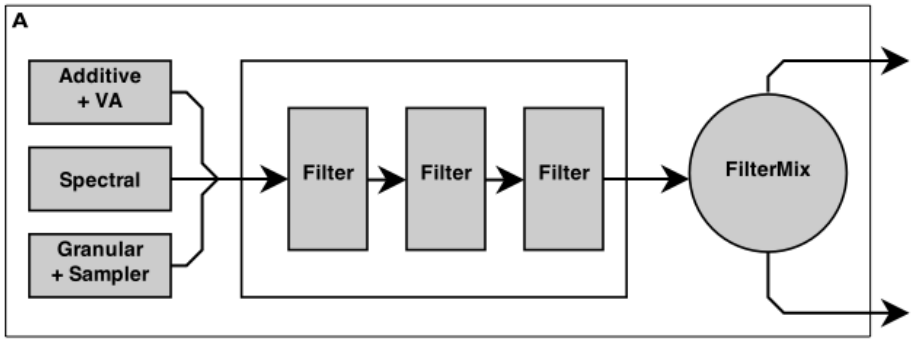
A Granular element, which can also be used for conventional sample playback (Sampler mode).

These can be toggled on or off independently within a single source, the only restriction being that when the Granular element is on, the Additive element can only be used in VA mode and the Spectral element can only be used as a noise source. To combine synthesis methods without facing this restriction, you can use multiple Sources.

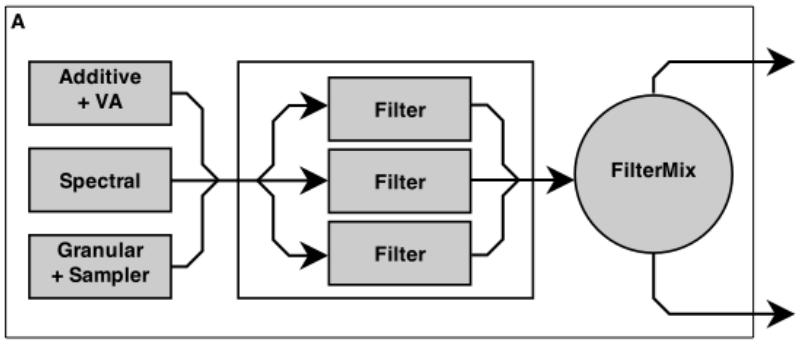
9.1.1 Source filters

The **Source filter** module provides three multi-mode filters, which can be configured either in **series** or in **parallel**. The Source filters allow you to filter each Source independently from the others. (Compare the Main Filters, which process a mixture of all four Sources.)

With the Source filters in **series** the signal passes through the three filters one after the other, like this:



With the Source filters in **parallel** the signal is split and fed through the three filters simultaneously, like this:



> **Note**: When the filter configuration is parallel and each filter is set to the 'Formant' type, a variety of Formant filter effects can be produced. See the example at the bottom of the Filter page for details.

9.1.2 FilterMix

The **FilterMix** control works like a kind of two-channel mixer (more accurately, a **crossfader**).

With the FilterMix control turned all the way to the left, all of the Source's output is fed to Main filter 1, and none of it to Main filter 2. (When you set the FilterMix knob this way, the parameter value display reads '100% F1 0% F2'.)

With the FilterMix control turned all the way to the right, all of the Source's output is fed to Main filter 2, and none of it to Main filter 1. (When you set the FilterMix knob this way, the parameter value display reads '0% F1 100% F2'.)

With the FilterMix control set half-way, equal amounts of signal are sent to Main filters 1 and 2. (When you set the FilterMix knob this way, the parameter value display reads '50% F1 50% F2'.)

9.1.3 Global page controls

At the top of the Global page's SOURCE section are five buttons: ALL, **A**, **B**, **C**, and **D**. When the **ALL** button is illuminated, the source master controls are displayed.

When the **A**, **B**, **C**, or **D** button is activated, the Source sub-page controls for the selected Source are displayed.

Each source includes a VU meter indicating the current level of output from the source to the Main Filter section. When the output level exceeds 0 dB (or 'clips' the input of the main filter section) the indicator will change colour to red. This is useful for quickly determining the cause of unwanted distortion in the overall output of Alchemy.

9.1.4 Source master controls

The Source master controls are a basic set of controls, duplicated for each of Alchemy's four Sources. These are a subset of the controls accessible from each Source sub-page.

On/off button (labelled '**A**', '**B**', '**C**', '**D**'). Toggles the selected source on or off.

Source content selection field. Displays the name of the current Source audio data. **Clicking** this field opens a pop-up menu with the following commands:

Load Audio - Quickly imports audio data (WAV, AIFF, or SFZ) into the Granular element, for playback in Granular or ordinary Sampler mode. Turns the Granular element on and the other elements off. If a SFZ file includes filter settings, configures a Source filter accordingly.

Load VA - Quickly loads waveform data into the Additive element, for playback in VA mode. Turns the Additive element on and the other elements off.

Import Audio - Opens a browser window for previewing sample data and importing it into the Granular, Additive, or Spectral elements. (See the Import page for details.)

Load Source and **Save Source** - Loads Source data from disk and writes it to disk. Uses special SRC file format. Loaded/saved data includes settings of all Source controls and a reference to loaded/imported audio data. (Source control ***modulations are not loaded/saved*** as part of the SRC file.)

Copy Source, Paste Source - Copies Source data to the clipboard and pastes it from the clipboard. Useful for duplicating the content and settings of one Source in another. Copied/pasted data includes settings of all Source controls and a reference to loaded/imported audio data. (Source control ***modulations are copied/pasted*** as part of the clipboard data.)

Clear Source - Reset the source to the default state

Randomise Source - Applies random variations to relevant Source controls, depending on which elements are active in the Source.

Load Additive/Spectral and **Save Additive/Spectral** – the commands are described separately in the Additive and Spectral data: AAZ and CSV files section.

Amp adjusts the level of the Source's output (-inf dB to 0 dB).

Tune adjusts the level of the Source's output, in semitone increments.

> Note that a **Fine Tune** control is available on each Source sub-page.

Pan positions the Source's output in the stereo field. (It works as a pan control if the source STEREO button is off, and as a left/right balance control if the source STEREO button is on.)

FiltMix controls the routing of the Source output to Main Filters 1 and/or 2, as described above.

The **MORPH** controls (pop-up menu, **X** and **Y** knobs) are described on their own page.

9.1.5 Source sub-page controls

Each of Alchemy's four sources has an **identical set of controls** for more in-depth editing:

Several of the Source sub-page controls are duplicates of those found on the Source master panel. These include the **On/off** button, the **Source content** selection field, and controls for **Amp**, **Coarse Tune** (corresponds to 'Tune' on

the master panel), **Pan**, and **FiltMix**. See details in the discussion of the Source master controls, above.

Each Source sub-page offers also offers several additional controls:

The **Loop mode** selection field contains a pop-up menu with five choices:

None — Ignores the loop start and end points, and plays the entire sound once without looping.

Continuous — Plays from the beginning, enters the loop region, loops continuously in a forward direction while a note is played, and goes on looping during the release stage.

Sustain —Plays from the beginning, enters the loop region, loops continuously while a note is played, and exits the loop region to play the remainder of the sound during the release stage.

Forward/Back —Like Continuous, but plays the loop region alternately forward and backward. If jumping from the end of the loop region back to its beginning produces an unwanted discontinuity, Forward/Back mode may give you more continuous results.

All —Ignores the loop start and end points and loops the entire sound continuously.

> **Note**: The **loop start** and **loop end** points can be edited in the Main view of the Source Edit page. The Additive element in VA mode, and the Spectral element in WHITENOISE mode, use raw oscillators and noise sources rather than loopable data, so elements using these types of synthesis are not affected by the Loop mode setting.

Solo button. When lit, isolates the Source by turning off all the other sources. Click again to restore the normal on/off status of other Sources.

Note. If you save a preset with one Source in Solo mode, the result is a preset in which that one Source is on (but no longer in Solo mode) and the other Sources are off.

Edit button. Gives you access to the Source Edit page.

The **KTRACK** selection field contains a pop-up menu with three choices:

Key+PBend — The pitch of the Source responds normally to MIDI Note data and PitchBend data.

Key — The pitch of the Source responds normally to MIDI Note data but does not respond to PitchBend data.

Off — The pitch of the Source does not respond to either MIDI Note data or PitchBend data.

> **Note** that the 'normal' response to PitchBend is set via the PitchBend Up and Down controls in Alchemy's Master section.

Stereo button. Toggles stereo mode. If stereo mode is **off** and a stereo file is loaded/imported, only the left channel will be played. When stereo mode is **on**, loaded/imported sounds are played in stereo and various manipulations such as panning individual oscillators in the Additive element and panning individual grains in the Granular element are possible. (See the Additive and Granular pages for details.)

Fine Tune adjusts the Source pitch in increments of one cent (one hundredth of a semitone).

Position. Determines the position within the audio data from which playback starts. (0% = very start of data, 100% = very end.) See ‘Modulating Position’ below for more information.

Stretch. In Additive, Spectral, or Granular mode, the Stretch setting determines the rate at which playback travels through the audio data. A setting of 100% represents the original playback rate. Higher settings (up to 500%) represent faster playback, while lower settings (down to 0%) represent slower playback.

- Playback remains at the normal pitch regardless of the rate of travel.
- Playback begins at Position and travels through the audio data on a path determined by the Loop mode. The Stretch setting determines the rate of this travel. (However, see 'Modulating Position' below for an alternative way to control the playback rate and path.)
- Setting Stretch to 0% ‘freezes’ playback at the position determined by Position.
- Stretch has no effect when the Granular element is set to ordinary Sampler mode.

9.1.6 Modulating Position

Position is a **modulation target**, which means you can control the path of playback through the audio data using any modulation source.

> **Note**: When the Granular mode is set to 'Sampler', the modulation value when Note-On is received determines the initial offset for the play position within the audio data. Beginning at that position, the rest of the sound plays in a normal manner, although looped as if the Loop mode were set to 'All'.

In Additive, Spectral, or Granular mode, Position may be **continuously** modulated i.e. forward or backwards, and at any rate including 0.

Try setting Position and Stretch to 0% and modulating Position with a unipolar LFO, an MSEG, or a Perform control linked to the modwheel.

When Position is modulated and Stretch has a value greater than zero, the playback path is determined by a combination of the modulation value (whenever this value changes) and the 'normal' path (travelled at a rate determined by Stretch whenever the modulation value is static).

> **Tip**: The example at the bottom of this page illustrates how to use modulation of the Position value to create a **tempo-synced loop**.

9.1.7 Elements and filters

A further four sub-pages are accessible from the bottom of each Source sub-page, which contain controls for the Source's elements and its Source filters. For information on the element controls, see the Additive, Virtual Analog (VA), Spectral, and Granular pages. The Source filters are described above. For details on filter types and filter controls, see the Main Filter page.

9.1.8 Example: Tempo-synced loops

Synchronising playback of looped audio with the host tempo is easy to achieve by modulating the Position parameter appropriately. This technique is possible with any synthesis method in Alchemy that permits continuous modulation of Position. In the example below, we use the Granular engine, but the same technique can be applied to the Additive and Spectral engines.

Initialise Alchemy by choosing the 'Clear' command in the Title Bar **FILE** menu.

Load (into Source A) a rhythmic or melodic sample that works well when looped. Try the following Factory sample:

Load Audio > Factory > Loops > DrumLoops > Cyborg-4bts.wav

The Load Audio command imports this audio file quickly into the Granular engine. You could also Import the file, using any import mode, and the technique presented here would work fine.

Now if you play and hold C3, the 'Cyborg-4bts' sample will **play in a looped fashion**. (Note that the Loop mode is already set to 'Continuous' by default.) If your host tempo happens to be 120 BPM, the loop will stay in tempo reasonably well — but suppose you want to sync the loop automatically to a different tempo set in your host.

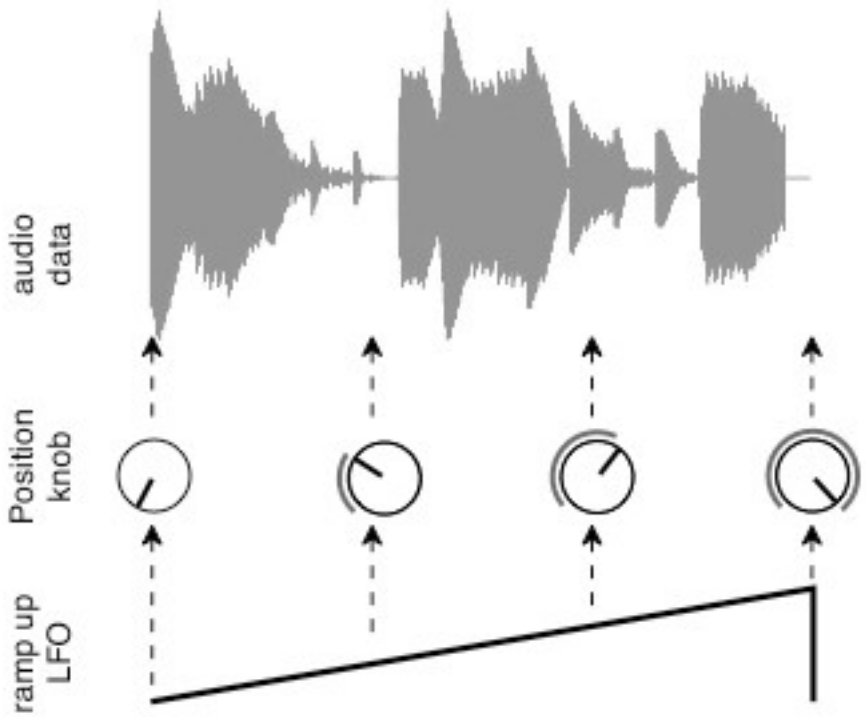
Turn the Source A Stretch knob down to its minimum value of 0%. Now if you play notes you'll find that **playback is frozen at the very beginning of the sample**. (All you'll hear is a low rumble, because all of the individual grains are being drawn from the low bass sound that begins the sample.)

Next we'll **assign a modulator to the Position parameter** in order to control a playback path through the audio data. First we want the Position to increase smoothly so that the entire sample plays back from beginning to end, then we want the Position to jump immediately back to the beginning, and finally we want to repeat this path so that the sound loops. The right modulator for this job is **an LFO with a ramp up shape**. Click the Position knob in order to access its mod rack in the Modulation section. In the first mod rack slot, choose **LFO 1**, which brings the LFO 1 controls into view on the right-hand side of the Modulation section. Finally, choose **RampUp** in the LFO Shape selection field and turn *off* the BIPOLAR button.

If you play and hold a note now, you'll find that the sample is looping as required, but the **playback speed** is much too fast. As its name suggests, the 'Cyborg-4bts' sample has a length of four beats. At its default rate, LFO 1 completes a cycle every beat. **Adjust the LFO Rate** to '4 beats'. Play and hold another note to confirm that the playback speed is now correct.

Finally, **adjust your host tempo** as you play additional notes to confirm that the loop is properly synchronised with the host tempo.

To summarise what's happening: a tempo-synced LFO with a ramp up shape is controlling the Position knob, which causes playback to scan through the audio data in a forward direction every four beats. The result is a tempo-synced loop. The diagram below depicts these relationships.



Note: it is not actually necessary to 'freeze' the playback by setting Stretch to 0%, as we've done in step 4, before modulating Position. The LFO shape we've chosen (ramp up) is constantly changing, without any static/flat portions; this means that it always controls play position, which makes the value of Stretch irrelevant.

To check this try double-clicking Stretch to restore its default 100% value. Then play a note: the looping behaviour remains the same.

However, if we were to use a different LFO shape with static/flat portions, setting Stretch to a non-zero value would cause play position to be controlled by a combination of the LFO shape and the normal play behaviour of the sample.

Try changing the LFO shape to 'RandHold' - the results may be more musically interesting if you also increase the LFO rate to '1/2 beats'. Play notes and compare the results you get with Stretch set to 0% versus 100%.

9.2 Morphing

The Morph X and Y controls determine how Alchemy's four Sources interact. There are two basic types of interaction.

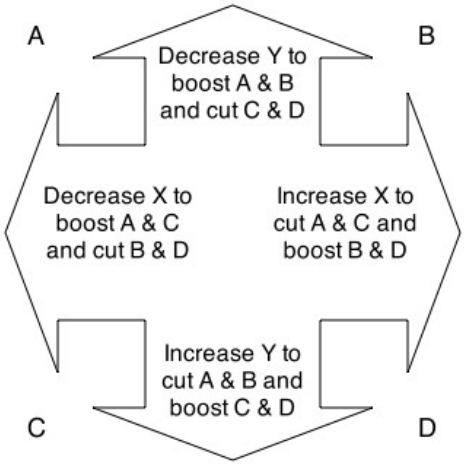
XFade. In an xfade (or 'crossfade'), sounds from all four Sources are played at once, and the X and Y knobs control the mix between them. This is equivalent to turning the Amp knobs in each Source up and down to get the desired mix.

> If you **xfade** from a Source with a *high* Coarse Tune setting to a Source with a *low* Coarse Tune setting, then the high Source will fade out as the low one fades in, and in the middle of the xfade you'll hear the high and low Sources simultaneously.

Morph. In a morph, a single sound is generated, and the parameters of that sound are determined by interpolating between the settings of all four Sources.

> If you morph from a Source with a *high* Coarse Tune setting to a Source with a *low* Coarse Tune setting, then you'll hear a single sound during the entire morph, and its tuning will fall smoothly from the high value to the low one.

The **Morph pop-up menu** offers a variety of xfade and morph modes.


A
B
Decrease Y to boost A & B and cut C & D
Decrease X to boost A & C and cut B & D
Increase X to cut A & C and boost B & D
Increase Y to cut A & B and boost C & D
C
D

xfade linear — The X knob crossfades from Source A to B to C to D as you increase it from 0% to 100%. (The Y knob does nothing in this mode.) This mode is useful for setting up crossfades based on Velocity or KeyFollow.

morph linear — This mode is like xfade linear, but all the parameters of the sound are morphed. Regions of each source sound bounded by corresponding Warp Markers are time-aligned in the morph — the Source Edit (Main) page provides a fuller explanation.

xfade xy — This is the **default morph/xfade mode**, and it crossfades between all four sources depending on the values of the X and Y knobs. X controls the mix levels of Source A & C versus B & D. Y controls the mix levels of Source A & B versus C & D.

morph xy — This mode is like xfade xy, but all the parameters of the sound are morphed. Regions of each source sound bounded by corresponding **Warp Markers** are time-aligned in the morph — the Source Edit (Main) page provides a fuller explanation.

> **Note**: that all of the remaining modes work similarly to morph xy, except that they provide control over the morph position of particular aspects of the sound. The settings you make in each of these more particular modes **coexist** with one another - for instance, if you first use the X and Y knobs in **add xy mode** (described below) to set values for the MorAddX and MorAddY parameters, then you can afterwards use the X and Y knobs in **spec xy mode** (described below) to adjust the MorSpecX and MorSpecY parameters without losing your MorAddX and MorAddY settings.

time xy — The X and Y knobs control the MorTimeX and MorTimeY parameters, which morph the timing of the sound, so if Source A has a short attack and Source B has a long attack, the length of the attack will vary as you change the X knob. In this mode, all the main Source parameters such as Amp, Pan, Tune, FilterMix,

Position, and Stretch are morphed as well. (The attack portion of each sound is determined by the positioning of **Warp Markers** in the Source Editor (Main) view.)

add xy — The X and Y knobs control the MorAddX and MorAddY parameters, which morph the Additive element between the settings of Sources A, B, C, and D.

spec xy — The X and Y knobs control the MorSpecX and MorSpecY parameters, which morph the Spectral element between the settings of Sources A, B, C, and D.

gran xy — The X and Y knobs control the MorGranX and MorGranY parameters, which morph the Granular element between the settings of Sources A, B, C, and D.

filt xy — The X and Y knobs control the MorFiltX and MorFiltY parameters, which morph the settings of the three Source Filter cutoff and resonance values between the settings of Sources A, B, C, and D. (The filter types and on/off status set in Source A remain in effect throughout the morph.)

> **See** the example at the bottom of this page for discussion of an easy way to work with several of these more particular morph modes simultaneously.

9.2.1 What parameters participate in a morph?

While most parameters can be morphed, parameters that are set via a pop-up menu cannot. For example, the values of Source Filter Cutoff and Resonance knobs can be morphed, but the filter types cannot. Similarly, the Additive Pitch, Amp, and Pan knobs can be morphed, but the profiles associated with these knobs - such as 'Harmonic' and 'OddEven' - cannot. Whenever a parameter cannot be morphed, the setting in Source A remains in effect throughout the morph.

9.2.2 Where did the 'Morph Square' go?

Users of Camel Audio's Cameleon 5000 additive synthesiser will already be familiar with the concept of morphing between four sources. In Cameleon 5000, these were called 'voices', and they were configured as the four corners of a Morph Square. Dragging a control point in this square gave users manual control over morphing, and complex automated morphs within the square could be created with the Morph Timeline.

In Alchemy, morphing between Sources A-D is controlled directly by the Morph X and Y knobs. But Alchemy offers equivalents to both the Morph Square and the Morph Timeline. Specifically, you can use one of the XY Pads in the Performance section like a **Morph Square** by assigning its X axis to modulate the Morph X knob and its Y axis to modulate the Morph Y knob. (See the example below for more details.) And you can use an XY MSEG in the Mod section like a **Morph Timeline** by assigning its X MSEG to modulate the Morph X knob and its Y MSEG to modulate the Morph Y knob. (See the example below for more details.)

9.2.3 Morphing Examples

Basic additive and spectral morphing

Initialise Alchemy by choosing the 'Clear' command in the Title Bar **FILE** menu.

Import a Factory Sample into each Source A-D, using **ADD+SPEC** mode. The following choices work well for the purposes of this example:

Source A: Woodwinds > Flute-Alto > Flute-Alto-C3

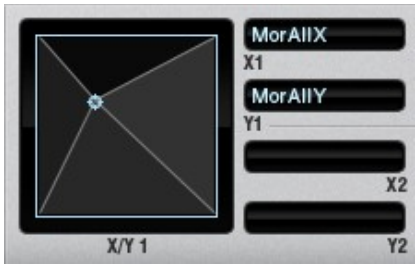
Source B: Keys > Piano-BabyG > Piano-BabyG-C3

Source C: Keys > EPiano-FM-Classic > EPiano-FM-Classic-C3

Source D: Woodwinds > Oboe-Vib > OboeB-C3

Go to each Source Spectral sub-page and double-click the **High Pass** knob; this reduces the High Pass setting and makes the Spectral component of the sound more prominent (which is useful for the purposes of this example). Then return to the **Source All** sub-page, which brings the **Morph X** and **Y** knobs into view.

Choose **morph xy** from the Morph mode selection field. Now you can use the Morph X and Y knobs to morph between Sources A-D.



Next, we'll assign an **XY Pad** (in the Perform section) to control the MorAllX and MorAllY parameters.

Set the **Morph X** knob to 0.0%; then right-click (control-click) it and choose Add Modulation > Perform > XYPad1X. Likewise, set the **Morph Y** knob to 0.0%; then right-click it and choose Add Modulation > Perform > XYPad1Y.

Now you can **morph between Sources A-D by dragging the control point** in XY Pad 1. Users of Camel Audio's Cameleon 5000 synthesiser will recognise that this result replicates Cameleon 5000's Morph Square in All mode.

Morphing different aspects of the sound

In the previous example — which is a starting point for this one — we replicated Cameleon 5000's Morph Square in 'All' mode. In Cameleon 5000, it is also possible to switch the Morph Square to 'Amplitude, 'Harmonics', or 'Noise' mode. Colour-coded control points for each of these modes are superimposed on Cameleon 5000's Morph Square, and you can set the position of each of these points separately by switching among the different modes. Alchemy's XY Pads are not designed to display multiple, colour-coded control points. In this next example, we'll explore an easy way to have simultaneous, independent control over the morphing of different aspects of a sound in Alchemy.

To begin, 'undo' the XY Pad assignment made in the final step of the previous example: right-click (control-click) XY Pad 1 and choose Delete Modulation X > MorAllX; then right-click XY Pad 2 and choose Delete Modulation Y > MorAllY.

Next, choose **time xy** from the Morph mode selection field. This gives you control over the MorTimeX and MorTimeY parameters and corresponds to Cameleon 5000's 'Amplitude' morph mode. Set the Morph X knob to 0.0%; then right-click (control-click) it and choose Add Modulation > Perform > Control1. Likewise, set the Morph Y knob to 0.0%; then right-click it and choose Add Modulation > Perform > Control5.

Similarly, choose **add xy** from the Morph mode selection field. This gives you control over the MorAddX and MorAddY parameters and corresponds to Cameleon 5000's Harmonics morph mode. Set the Morph X knob to 0.0%; then right-click (control-click) it and choose Add Modulation > Perform > Control2. Likewise, set the Morph Y knob to 0.0%; then right-click it and choose Add Modulation > Perform > Control6.

Finally, choose **spec xy** from the Morph mode selection field. This gives you control over the MorSpecX and MorSpecY parameters and corresponds to Cameleon 5000's 'Noise' morph mode. Set the Morph X knob to 0.0%; then right-click (control-click) it and choose Add Modulation > Perform > Control3. Likewise, set the Morph Y knob to 0.0%; then right-click it and choose Add Modulation > Perform > Control7.

This result gives you six Performance knobs that you can use to control morphing of the additive, spectral, and envelope-timing aspects of Sources A-D. (Try using MIDI learn to map these knobs to your controller hardware!)

Example continued: MSEG-driven morphs

Let's take our example one step further and construct the equivalent of Cameleon 5000's 'Morph Timeline', which is a powerful tool for creating complex, rhythmic morphs. You can use the results of either of the preceding examples— 'basic additive and spectral morphing' or 'morphing different aspects' - as a starting point for the steps below.

Undo any Performance control assignments: right-click (control-click) all the active Performance knobs and use the 'Delete Modulation' options to find and delete knob modulations; or right-click (control-click) the active XY Pad and use the 'Delete Modulation X' and 'Delete Modulation Y' options to find and delete pad modulations.

If necessary, restore the morph mode to **morph xy**. Right-click (control-click) the **Morph X** knob and choose Add Modulation > MSEG > MSEG 1; then right-click the **Morph Y** knob and choose Add Modulation > MSEG > MSEG 2.

Click the XY MSEG button at the top right of the Modulation section to bring the **XY MSEG controls** into view. Note that MSEG 1 and MSEG 2, which we have already assigned to modulate the morph knobs, both participate in the default XY MSEG configuration. (There is a dedicated page describing how to work with the XY MSEG controls; for this example, we'll simply load a preset into the default configuration.)

Load an XY MSEG preset: click the FILE button, and choose 'Load' from the pop-up menu. From the factory 'MSEG' folder, choose Rhythmic-Simple > RSquare 1_4X.mse. The chosen file is loaded into MSEG 1, and the associated file RSquare 1_4Y.mse is automatically loaded into MSEG 2 as well.

Play a few held notes to confirm that the sound now morphs rhythmically between the settings of Sources A-D.

You may find that Alchemy's faster envelope response leads to unwanted 'clicks' each time the morph position jumps to a different corner of the square. Let's make one refinement to eliminate any artefacts of this type. Click the Morph X knob to bring the MorAllX parameter's mod rack into view. (You'll see MSEG 1 is already assigned as a modulator.) Find the **Smooth** knob above the mod rack and raise it slightly, to about 1.0%. Likewise, click the Morph Y knob to bring the MorAllY parameter's mod rack into view, and again raise the Smooth knob to about 1.0%.

9.3 Importing Audio

Choosing **Import Audio** from the pop-up menu in a Source content field opens the **Import Browser** window.

The central portion of the window provides a simple file browser in which you can select WAV, AIFF and SFZ format sound files for importing.

The **AUTOPREV** button toggles automatic previewing on and off. With **AUTOPREV** activated, you can hear a preview of a sound file simply by clicking its name. Sound files can be mono or stereo, 8-, 16- or 32-bit, at any sample rate (although rates higher than 44.1 KHz are not recommended, since they don't yield any significant improvements in quality).

You can import sound files to any of Alchemy's various elements: to granular, additive, or spectral, or to a combination of additive and spectral. To choose one of these import modes, use the GRANULAR, ADDITIVE, ADD+SPEC, and SPECTRAL buttons across the top of the Import window. Importing to the additive and/or spectral elements requires Alchemy to perform a special analysis of the file, which may take a moment.

> **Tip**: place shortcuts (Windows), aliases or symbolic links (Mac) to frequently used sample folders elsewhere on your system, in the Alchemy/Samples folder, for quick access to your sample collection via the Import Audio browser.

9.3.1 Importing multiple files

Multiple files may be imported to a single source by holding down the **Shift** key when selecting files in the Import Audio window. With multiple files selected, releasing the shift key and selecting a new file will reset the selection to a single file. Clicking the **Import** button will import the selected files using the current analysis settings.

> **Note**: it is only possible to select WAV and AIF files for multi-file import.

If the note name is included in the file name, samples will be mapped to the appropriate keyboard zones for all analysis types. When importing using additive or spectral analysis, samples without a note name will be analysed to determine pitch information. The pitch information is used to determine the most appropriate zone for keyboard mapping. Samples will be mapped to the highest key of the zone, and pitched down for the remaining notes in the zone.

When no pitch information or note names are available samples will be mapped evenly across the keyboard zones based on the order in which the files were selected. The pitch key centre is set to the middle of each zone.

A description of each import mode follows, with comments on the relevant options.

9.3.2 Granular Import

Importing to the granular element is good for drums loops, percussive sounds, and any sound to which you want to apply special granular effects. You can also use granular import and then switch the Granular element to its ordinary 'Sampler' mode, which is the best choice when you want to play a sound file efficiently in its original form, transposed across the keyboard.

RootNote determines which MIDI note will play the sound file at its original transposition. If the filename has a pitch appended to it ('MyFile C2', 'YourFile-E3', 'OurFile_F#4'), Alchemy will automatically set the RootNote parameter to match.

The **Range** and **TimeFreqAcc** parameters do not apply to Granular import.

> **Note**: If the sound quality is not what you expected, check to make sure the Granular element is switched to 'Sampler' mode, unless you intend to perform granular-specific manipulations such as time stretching/shuffling.

9.3.3 Additive Import

Importing to the Additive element allows for the most detailed manipulation of sounds, and is especially good for sound files that represent single notes (rather than chords or more complex sounds and textures).

Good results with additive importing depend on an appropriate choice of **RootNote**. If the filename has a pitch appended to it, Alchemy will automatically set the RootNote parameter to match; otherwise, it will attempt to determine a suitable RootNote based on a preliminary analysis of the waveform.

> **Note**: If a sound fails to import as well as you think it ought to, you may be able to get improved results by shifting the RootNote setting up or down an octave.

The **Range** setting is also significant. For sound files that represent single notes (rather than melodies, chords, or more complex sounds and textures), choose 'SingleNote'. For most other sounds, choose 'OneOctave'. Again, Alchemy will attempt to determine an appropriate setting based on a preliminary analysis of the waveform.

TimeFreqAcc determines whether 'time accuracy' or 'frequency accuracy' is prioritised during the import process. In most cases you can leave this at the default ('GoodTime'), but for some material different settings may produce better results. (For example, to analyse speech, you should probably choose 'BestFreq'.)

9.3.4 Add+Spec Import

When a purely Additive import fails to capture the noisy components of a sound (such as the hammer strike of a piano or the breath noise of a flute), a combination of Additive and Spectral importing may give the best results.

After an Add+Spec import, try deactivating the Source's Additive element and listening to the Spectral element on its own. This will allow you to hear which parts of the sound failed to be reproduced by the Additive element, and should give you an idea what kinds of sounds are better-suited to the different analysis methods.

By default, the Spectral component of an Add+Spec import is played with the Spectral element set to 'Noise-Resynth' mode. Playing the spectral data back in 'Resynth' mode produces a markedly different effect, which you may sometimes prefer.

By default, the Spectral element's internal highpass control is set relatively high in order to exclude frequencies that would compete with those produced by the Additive element. If you find the default contribution from the Spectral element to be too subtle, you can make it more prominent in many cases by setting the highpass control to a lower value.

> **Note**: If a sound fails to import as well as you think it ought to, you may be able to get improved results by shifting the RootNote setting up or down an octave.

9.3.5 Spectral Import

Importing to the Spectral element allows effective manipulation of polyphonic sounds (chords, drum loops, and other complex sounds and textures that don't analyse well via Additive).

Set **RootNote** to determine which MIDI note will play the resynthesised sound at its original transposition. If the filename has a pitch appended to it, Alchemy will automatically set the RootNote parameter to match.

The **TimeFreqAcc** parameter can also be adjusted to suit different types of source material (see details in the discussion of Additive import, above).

The **Range** parameter **does not** apply to Spectral import.

9.3.6 'Import Audio' versus 'Load Audio'

The 'Load Audio' and 'Import Audio' commands available in the Source content field's pop-up menu both bring audio files into Alchemy.

Load Audio can be used load a single WAV, AIFF, or SFZ file rapidly into the Granular element for playback in Granular or Sampler mode. You can also load files by clicking the Source content field's forward and back arrows. If an SFZ file is loaded the previous/next buttons will skip over individual samples and load the previous or next SFZ file. This is a convenient way of auditioning different instruments. If a single sample is loaded, SFZ files will be skipped in favour of individual audio files.

Import Audio is for when you want Alchemy to analyse and resynthesise a sound file, to quickly import multiple files to a single source, or when you want to preview sounds before loading them. When a sound has been imported with additive or spectral analysis, the resynthesis data is stored in an '.aaz' file in the same folder where the preset is saved. The first part of the .aaz filename is the same as the preset name, ensuring that the preset and its associated .aaz files will be found side by side in an alphabetical directory.

> **Note**: It is recommended that you *avoid* renaming .aaz files externally (i.e. from your operating system); Alchemy may fail to locate and load these files unless their original names are preserved. While renaming a whole preset externally is less risky, it may lead to duplication of .aaz files to reflect the new name. In general, it's best to rename presets by using Alchemy's SAVEAS function. Once a preset is successfully saved under its new name, you can safely discard the old preset (*.acp) file.

9.3.7 Supported SFZ opcodes

Alchemy is capable of importing multi-zone sample data in SFZ format. SFZ is a non-proprietary file format (described at http://www.cakewalk.com/DevXchange/sfz.asp).

Alchemy recognises the **<region>** and **<group>** headers and the following opcodes:

```
sample
pitch_keycenter
key
lokey
hikey
lovel
hivel
loop_mode
cutoff
fil_veltrack
tune
volume
pan
seq_position
trigger
sw_last
default_path
```

> **Note**: Alchemy prefers SFZ files that contain note numbers (e.g. hikey=54) rather than note names (e.g. hikey=f#3), to always remain compatible with both MIDI note 48 and 60 middle C, note number conventions, and to avoid any octave jumps around middle C.

You can create your own SFZ files for use in Alchemy by using various third-party software tools. SFZ definition files are simple text files, so another approach to assembling your own SFZ files is to place a copy of an existing definition file in a folder along with the new samples you want to use, and then to edit the definition file - in a text editor such as Notepad (PC) or TextEdit (Mac) - so that it references your new samples.

See the SFZ section towards the end of this manual for detailed information about creating your own SFZ files.

9.4 A Limit on the Size of Imported Sample Data

Alchemy supports an unlimited number of zones when importing from SFZ files. However, the default settings of the Import Audio browser will limit the amount of memory used by intelligently reducing the number of key switches, velocity layers and round-robin samples referenced by the selected SFZ file. The amount of memory required by an SFZ is displayed in the Memory field next to the 'Cancel' button. You may override the choices by selecting new values for 'Keyswitch', 'Max Velocities' and 'Max Round Robins'; the memory field will be updated accordingly so you can estimate the resources required for the import.

The default limit used by Alchemy is around 100MB. SFZ files referencing more than 100MB will cause a reduction in the number of keyswitches, velocity layers and round robin layers until the limit is reached. You may change the default limit by modifying the 'DefaultMaxSfzSize'; parameter in the Alchemy config file; the default size is specified in MB.

Alchemy doesn’t directly apply any limits on the length of sounds that you can import into its synthesis engines. However, Alchemy will fail to import sounds if it runs out of memory. For spectral and additive analysis, the amount of memory required is several times the actual size of the WAV or AIFF file. A general rule of thumb is that if you have at least 1GB of free memory, you should be able to import sounds up to five minutes in length with any analysis method without problems.

9.5 A Limit on the Size of Sample Folders

In any single folder within Alchemy/Samples, no more than 1024 samples and folders are permitted. (This limit allows Alchemy to construct menus safely when you choose the "Load Audio' popup command in a Source content field.) If you really want to have, say, 2000 samples in a folder such as Samples/User/Bass, you can organise them into subfolders within that folder. For instance, Samples/User/Bass might contain 500 samples plus five additional folders, each of which contains 300 more samples.

10 Additive Synthesis

10.1 A brief introduction to additive synthesis...

Additive synthesis produces complex sounds by summing together a number of simpler ones, known as **partials**. Typically, each partial is a pure sine wave tuned to be a multiple of some fundamental frequency *f*, so that you have partials at *f*, 2*f*, 3*f*, and so on. Additive synthesis puts the mathematician Joseph Fourier's theory, which states that every static waveform can be decomposed into a sum of sine-wave partials, into practice.

Of course most musically useful sounds involve waveforms that **change over time**; producing these sounds with additive synthesis requires changing the level of each partial accordingly. Another important additive technique involves tuning partials higher or lower than their harmonic positions. These **inharmonic partials** are characteristic of the patterns produced when stiff materials such as metal and glass vibrate.

Two features make the additive engine in Alchemy so powerful. First, it offers an unusually **large number of partials** – as many as 600 in each of its four Sources. And second, the amplitude, pitch, and pan of each partial can be specified across an **unlimited number of breakpoints**. That adds up to a potentially huge amount of data, but Alchemy can manage the details for you thanks to its **resynthesis** capabilities.

Resynthesis is a two-step process. First, an audio file (in a standard format such as WAV or AIFF) is **analysed** to determine its partials and their behaviour over time. And second, a replica of the original sound is **synthesised** using the results of the analysis.

Resynthesis in Alchemy produces results that are remarkably close to the sound of the original audio file. But then the real fun begins: because you now have independent control over the amplitude, pitch, pan, and phase of each partial, you can manipulate sounds in ways that would be impossible to achieve with a conventional sampler.

Alchemy enables you to perform these manipulations in real-time using a variety of modulation possibilities, described below. For precision work, you can also make use of Alchemy's dedicated Additive Editor, which allows you to work at varying levels of detail, right down to individual partial values and breakpoints.

10.2 ADD and VA mode buttons

The Additive element in each of Alchemy's four Sources can operate in one of two modes. The ADD button on the left of the Additive sub-page puts the element in **Additive** mode, while the adjacent VA button puts it in **Virtual Analog** mode. (Turning on one mode turns the other off.) Importing additive data, or creating it from scratch in the Additive Editor, automatically sets the mode to Additive.

Controls on the Additive sub-page apply to both VA and Additive mode. The descriptions below explain how these controls work in Additive mode. The effect of the same controls in VA mode is explained on the Virtual Analog (VA) page.

10.3 Using the Additive sub-page controls in ADD mode

Wave pop-up menu. Selects the **waveform** used to play each partial. In ADD mode, the 'Sine' waveform in the 'Basic' category is nearly always preferred, as it allows more accurate resynthesis and requires considerably less CPU load. When the total number of partials is very small, however, other choices of waveform may give you richer and more interesting results.

Vol. Adjusts the **volume** of the Additive element, independent of other elements in the same source.

Sym. Controls the **symmetry** of the waveform used to play each partial. At the default value of 50%, this control has no effect. At higher or lower values, one half of the waveform is stretched while the other half is compressed. Modulating this parameter produces pulsewidth modulation. To minimise CPU usage, however, leave this control at 50% and avoid modulating it.

PVar. Use this control to reduce the amount of **pitch variation** present in the additive data. At 100%, the original pitch variation is applied in full; at 0%, the pitch of each partial is fixed at its harmonic value. (If you're familiar with Cameleon 5000, keep in mind that 'PVar' in Alchemy is equivalent to 'Harm' in Cameleon 5000.)

> **Note** that pitch offsets resulting from the Pitch knob (described below) are not affected by the PVar setting.

NOsc. Use this control to reduce the **number of oscillators** used to play the additive data (and hence the number of higher partials that are heard). Using fewer oscillators conserves processing power, and you can often reduce NOsc without noticeably changing the resulting sound. More drastic reductions in the NOsc value produce an effect comparable to lowpass filtering. (Try modulating this parameter with an AHDSR.)

> **Note** that the additive engine processes partials in groups of four, so *setting NOsc to a multiple of four* gives you the best trade-off between high-end detail and CPU efficiency.

Pitch knob and pop-up menu. The Pitch knob applies a variable offset to each partial pitch. The nature of this offset depends on the setting chosen in the Pitch pop-up menu directly below the knob:

Harmonic - this setting is the default in ADD mode and is the most useful choice in most situations. When Pitch = 0%, all partials are tuned precisely in unison at the fundamental frequency. When Pitch = 50%, all partials have their normal harmonic tuning. When Pitch = 100%, the partial tunings are stretched significantly. (If you're interested in the math behind this stretching: partial *n* is tuned to the (2*n* - 1)th harmonic.)

Unison – this setting is mainly intended for VA mode and has limited use in ADD mode. (See the description on the Virtual Analog (VA) page for details.)

Amp knob and pop-up menu. The Amp knob applies a variable offset to each partial pitch. The nature of this offset depends on the setting chosen in the Amp pop-up menu directly below the knob:

Brightness- set Amp above 50% to boost the brighter (higher) partials, or below 50% to boost the darker (lower) ones.

Fifths - set Amp above 50% to boost the partials that are a fifth, plus one or more octaves, above the fundamental, or below 50% to boost the other partials.

FiveMul - set Amp above 50% to boost every fifth partial (5, 10, 15, 20, …) and cut the others, or below 50% to cut every fifth partial and boost the others.

Fundamental - set Amp above 50% to boost the fundamental and cut the other partials, or below 50% to cut the fundamental and boost the other partials.

Octaves - set Amp above 50% to boost the partials that are one or more octaves above the fundamental, or below 50% to boost the other partials.

OddEven - set Amp above 50% to boost the even partials and cut the odd ones, or below 50% to do the reverse. Boosting the odd partials creates a 'hollow' tone resembling the sound of a square wave.

Additional options in the Amp pop-up menu give you access to the **Additive Formant Filter**, which is described in more detail below.

Pan knob and pop-up menu. The Pan knob applies a variable offset to each partial's stereo position. At 50%, all partials are centred. Values lower than 50% pan some partials left and other partials right; values greater than 50% pan partials in the opposite direction. The nature of these offsets depends on the setting chosen in the Pan pop-up menu directly below the knob:

Brightness - pans darker partials to one side and brighter partials to the other.

OddEven - pans odd partials to one side and brighter partials to the other.

Unison - pans partials evenly across the stereo field.

> Note that you won't hear the effect of Pan unless the Source's **Stereo** button is engaged.

10.4 Using the Additive Formant Filter

The Additive Formant Filter produces complex filtering effects by adjusting the balance among all the oscillators that are summed together to play an additive sound. In other words, it changes the levels of partials relative to one another.

> **Note**: Because the Additive Formant Filter depends on changing the balance among additive partials, it can be used to filter a combination of many oscillators summed together to produce a complex spectrum - a typical *additive* configuration - but it cannot be used to filter a single oscillator with a complex spectrum - a typical *VA* configuration. To create formant-filter effects with Alchemy in VA mode, the best approach normally involves using the Source Filters in a parallel configuration (click for more details on this technique).

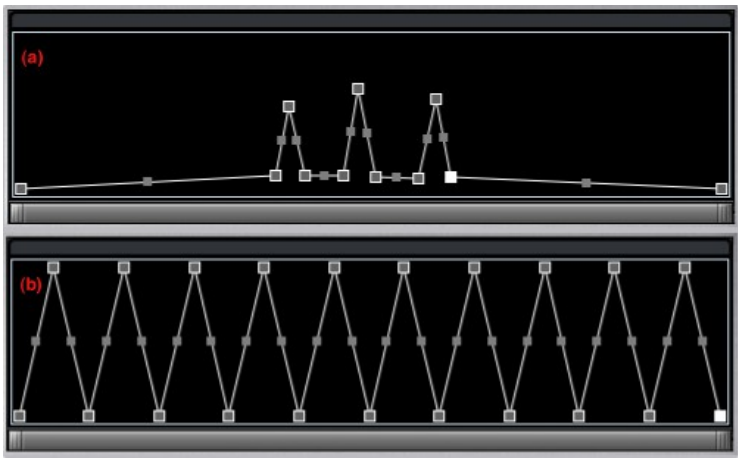
The Additive Formant Filter can be applied to the Additive Element of any Source. To access it, click the Amp pop-up menu, scroll to 'Formant ModMap', and choose 'Fmt 1' (or create a new one). When you choose Fmt <N>, Alchemy will interpret the Nth ModMap as a description of how to boost/cut various frequencies: the horizontal (x) axis represents frequency, while the vertical (y) axis represents amplitude. When the Amp knob is set to its default value of 50%, x values translate to frequencies approximately as shown in the following table. The entire formant-filter shape can be shifted upwards or downwards as much as five octaves when the Amp knob is set to values above or below 50%.

17	34	69	138	275	550	1100	2200	4400	8800	17600
0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%

ModMap x-value

Approximate filter frequency in Hz, when Amp knob is set to 50%

To alter the characteristics of the formant filter Fmt <N>, simply edit the shape of ModMap N. For instance, here are ModMap shapes that produce (a) a filter with three prominent midrange formants, and (b) a steep comb filter:



10.4.1 Example: Manipulating a resynthesised voice

Initialise Alchemy using the Title Bar and selecting **FILE** > Clear.

Click the 'A' button to view the **Source A** sub-page and choose **Import Audio**.

In the left-hand half of the Import dialog, select the sample Factory > Vocals > Caroline-CamelAudio.wav; then set the import mode to **ADD+SPEC** at the top right, and click **IMPORT** at the bottom right.

Now you can play a resynthesised version of the 'Caroline-CamelAudio' sample - play F-sharp 3 to hear the original pitch.

Try the following manipulations:

Reduce NOsc from 32 to about 4 and note how the voice becomes darker and less distinct. (Reset NOsc to 32.)

Stretching and compressing the tuning of the partials by adjusting the Pitch knob. Because Caroline sings clear pitches, the original spectrum is harmonic; when stretched or compressed, the spectrum becomes inharmonic at most settings, giving it a glassy or metallic quality. A setting of 25% gives the effect of a deep male voice. (Reset Pitch to 50%.)

Explore the Amp and Pan knobs as well. (Turn on the Source STEREO button to hear the effect of Pan.)

Finally, let's turn 'Caroline-CamelAudio' into a pad sound.

Set PVar to 0% in order to provide a steady pitch.

Slow down the playback by setting Stretch = 30% at the top of the Source A panel. Turn on the Source STEREO button.

Now set Pitch = 25% (using the 'Harmonic' profile), Amp = 0% (using the 'OddEven' profile), and Pan = 100% (using the 'Unison' profile).

Refine this result by adjusting the rest of the signal path. Explore the Main Filters. Try adding a ModFX or Delay module in the Effects rack.

10.5 Advanced: Creating your own Pitch, Amp, and Pan profiles

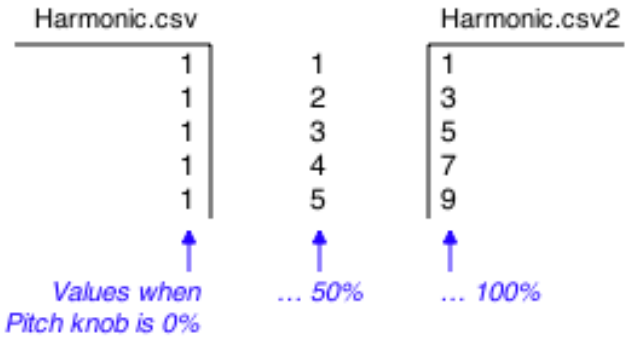
The preset profiles for Oscillator Pitch (Unison, Harmonic), Amp (Brightness, OddEven, etc.), and Pan (Unison, OddEven, etc.) are defined by special files stored in Alchemy/Libraries, where you will find subfolders named OscillatorPitch, OscillatorAmp, and OscillatorPan. These profile definitions are formatted as standard CSV files, which makes it easy to create your own definitions by using a text editor or spreadsheet, or by writing your own computer program that stores its output in a text file.

Each profile definition requires a pair of files with identical base names and different extensions: the profile corresponding to a knob value of 0% has the extension .csv, while the profile corresponding to a knob value of 100% has the extension .csv2. You may find it informative to study an existing pair, such as 'Harmonic.csv' and 'Harmonic.csv2' in the OscillatorPitch folder. Or read on for a more detailed explanation.

Each row of a profile definition holds just one value, corresponding to the pitch, amp, or pan value of a single partial. The first row applies to the first partial, the second row to the second partial, and so on. Since Alchemy generates up to 600 partials, an exhaustive profile definition should consist of 600 rows. If you are creating a new profile for a specific purpose, and you know you won't use more than, say, 60 partials, then there's no need to continue the profile definition beyond 60 rows. (If you use more oscillators than there are values in the profile definition, the surplus oscillators retain their previously set values.)

In a profile definition, pitch values are expressed as multiples of the fundamental frequency. (So the values '1, 1, 1, …' represent an exact unison; the values '1, 0.99, 1.01, …' represent a detuned unison; the values '1, 2, 3, …' represent the standard harmonic series; and so on.) Amp and Pan values are expressed on a scale from 0 to 1. (In the case of Pan values, this means from fully left to fully right in the stereo field).

Alchemy determines actual oscillator values by interpolating between the .csv and the .csv2 files, based on the position of the Pitch, Amp, or Pan knob. The details in the case of the 'Harmonic' Pitch profile are pictured below.



Let's create a new Amp profile that will enable us to cut or boost the levels of the prime-number partials relative to the others. For simplicity, we'll extend our profile only through the first 60 partials; constructing files of this length by hand in a text editor is a manageable task. When the Amp profile is set to our new 'Primes' definition, turning the Amp knob fully left should set the amplitudes of all prime-number partials to 0, and all nonprime-number partials to 1 (this tells us how 'Primes.csv' should be constructed), and turning the Amp knob fully right should do the reverse (this tells us how 'Primes.csv2' should be constructed).

The prime numbers up through 60 are: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59. So creating the 'Primes' profile involves the following steps:

Create a plain text file with one value per line, that value being '0' for the prime-number lines (2nd, 3rd, 5th, 7th, 11th, …) and '1' for the nonprime-number lines (1st, 4th, 6th, 8th, 9th, …); then save this file in Alchemy/Libraries/OscillatorAmp with the name 'Primes.csv'.

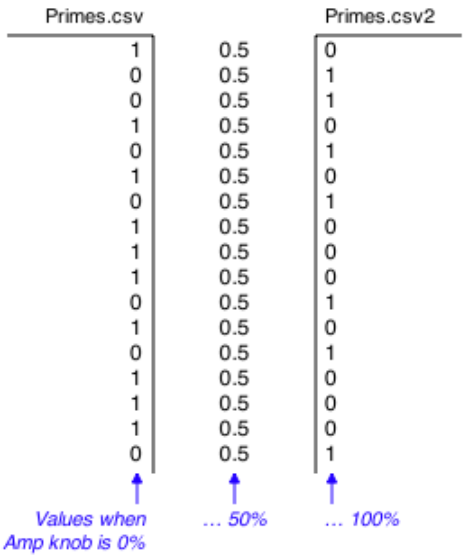
Create a second plain text file with one value per line, that value being ‘1’ for the prime-number lines and '0' for the nonprime-number lines; then save this file in the Alchemy data folder:

Alchemy/Libraries/OscillatorAmp

with the name 'Primes.csv2'.

The next time you load Alchemy in your host, the 'Primes' Amp profile should be listed along with the factory profiles. Give it a try!

See the illustration on the right for how the 'Primes' profile works (going as far as the 17th partial).



11 Virtual Analog (VA)

The **VA** button on each Source's Additive element sub-page gives you access to Alchemy's 'Virtual Analog' mode. In this mode, you can use the additive engine as a bank of raw oscillators such as you might find in a classic analog synthesiser.

The **controls** in VA mode are the same as in ADD mode. The Additive page gives you detailed descriptions of each control. The discussion below focuses on using these controls to perform various analog-synthesis tasks.

When you **initialise** Alchemy, using the 'Clear' command in the Title Bar **FILE** menu, the Source A Additive element is turned on and switched into VA mode. By default, a single oscillator is active and the Saw wave is selected.

11.1 VA techniques in Alchemy

To select the oscillator waveform, use the **Wave** pop-up menu. The 'Basic' category offers Saw, Sine, Square, and Triangle waves. A large variety of more specialised waves is also provided. You can browse through the available waveforms using the forward and back buttons, or make a direct selection from the pop-up menu.

To adjust the pulsewidth, use the **Symmetry** control. A setting of 50% preserves the original waveform, while higher or lower settings stretch one half of the wave and compress the other half. Starting with a Square wave gives you a classic variable-width pulse wave, but you can adjust the symmetry of all waveforms in Alchemy. Symmetry values as low as 5% and as high as 95% are typically useful; at the most extreme settings, the sound may become very thin or unstable.

For pulsewidth modulation (PWM), **modulate the Symmetry parameter** with an LFO, AHDSR, or other modulator.

To control oscillator phase, use the **PVar** knob. Values from 0% to 99.9% set a consistent starting position for the oscillator. When PVar = 100%, the starting phase of the waveform varies randomly per note.

> **Note**: the PVar knob has a different meaning in ADD versus VA mode. In VA mode, it provides control over phase. In ADD mode, it controls the depth of response to partial pitch variations in the additive data.

Set the number of oscillators using the **NOsc** knob. (The Pitch knob, described below, provides unison with variable detuning for each additional oscillator.) Alchemy lets you stack up to 600 oscillators per source (!) — but the high end of this range is really intended to provide sufficient sine wave partials for detailed additive resynthesis. In VA mode, typical settings of NOsc range from 1 for single-oscillator sounds, up to 20 or so for 'supersaw' and related effects. Higher settings of NOsc consume more CPU.

To control the amount of detuning (with NOsc = 2 or more), choose **Unison** in the **Pitch** pop-up menu and adjust the **Pitch** knob. A value of 0% tunes the oscillators to a perfect unison, while 100% provides the maximum amount of detuning.

> **Tip**: to create a punchy VA bass sound using only a single source set the NOsc control to '2' oscillators and PVar to value of 25% or 75% for fixed phase at note on.
>
> Set the Pitch knob to 3% or lower to reduce the amount of beating between the two oscillators.

To spread unison oscillators across the stereo field, choose **Unison** in the **Pan** pop-up menu and adjust the **Pan** knob. A value of 50% keeps all the oscillators centred, while higher or lower values distribute the oscillators from left to right.

> Note that you won't hear the effect of Pan unless the Source's **Stereo** button is engaged.

For white and filtered noise, use the Spectral element in WHITENOISE mode. (See the Spectral page for details.)

11.1.1 Example: Creating a 'supersaw' effect

Initialise Alchemy by choosing the 'Clear' command in the Title Bar **FILE** menu. This puts Source A into VA mode.

If the Source A **Additive sub-page** is not already in view, click the 'A' button and then click the Source A ADDITIVE button. **VA** mode will be enabled by default.

The **Saw wave** should already be selected in the Wave pop-up menu.

Increase **NOsc** to about 10. You may also wish to increase the value of **Pitch** above 50% for more extreme detuning.

Finally, try a couple of refinements:

For **modwheel control of the filter cutoff**, go to the Perform sub-page and choose 'Auto Assign All' from the FILE menu at the far right. Filter cutoff is now mapped to Perform Control 3, so choose 'Control3' in the ModWheel pop-up menu.

Experiment with **various effects modules** in the Effects rack. (Compressor and Camel Reverb are good ones to try.)

12 Spectral Synthesis

12.1 A brief introduction to spectral synthesis...

Spectral synthesis and additive synthesis represent two different ways of creating complex sounds by summing together a number of simpler components. In additive synthesis (which you can read about here), each component is a sine wave partial tuned in relation to a harmonic series. In spectral synthesis, the audible spectrum is divided into a large number of 'spectral bins', and different sounds are represented as different distributions of energy across these bins. Normally, Alchemy plays spectral data by summing together a number of sine waves centred in each bin; it can also fill each bin with filtered noise.

> **Note**: This process uses a model known as a *phase vocoder*. We use the term 'spectral synthesis' to avoid confusion: Alchemy does *not* provide a conventional 'vocoder' and is not designed to impose the spectral properties of one signal on another in real time.

Alchemy lets you **resynthesise** existing audio files (WAV or AIFF) using spectral techniques as well as additive ones. Your choice may depend on the type of material: additive resynthesis is often preferred for sounds that project a single clear pitch, while spectral resynthesis is well suited to chords and noisy sounds. Alchemy can also perform resynthesis using a *combination* of the two techniques, which is useful for sounds that combine a clear pitch with a noisy component (e.g. the hammer strike of a piano, or the breath noise of a flute).

See the Import page for instructions on how to perform resynthesis.

Spectral resynthesis allows you to **manipulate your source material** in unique ways. You can switch between resynthesis using sine waves and resynthesis using filtered noise — this can transform speaking into whispering. And for detailed work, you can edit spectral data by applying various painting tools in Alchemy's dedicated Spectral Editor.

Unlike the other elements in Alchemy, the Spectral element generates a **mono (rather than stereo) signal**, so the Source STEREO button has no effect on the Spectral element signal. If you import a stereo file using spectral resynthesis, the left channel will be used and the right channel will be discarded.

12.2 Modes of operation

The Spectral element in each source can operate in one of three different modes, selectable via the three Spectral Mode buttons at the left of the Spectral sub-page:

WHITENOISE produces steady white noise.

RESYNTH plays the current spectral data (from resynthesis or painted directly into the Spectral Editor) by summing sine waves.

NOISE-RESYNTH plays the current spectral data (from resynthesis or painted directly into the Spectral Editor) by summing bands of filtered noise. The result is still more or less 'pitched', but with a distinctive noisy/phasey quality.

12.3 Controls

KEYTRACK button. Determines whether or not the pitch of the Spectral element tracks the keyboard in RESYNTH and NOISE-RESYNTH modes. For key-tracking to occur, the main Source KTRACK field must also be set to 'Key+PBend' or 'Key'.

Volume. Adjusts the volume of the Spectral element, independent of other elements in the same source.

High Pass. Sets the frequency below which spectral bands are eliminated (acts like a steep highpass filter).

Low Pass. Sets the frequency above which spectral bands are eliminated (acts like a steep lowpass filter).

Amp knob and pop-up menu. Provides control over the Spectral Formant Filter, described in further detail below.

12.4 Using the Spectral Formant Filter

The Spectral Formant Filter produces complex filtering effects by adjusting the balance among all the components that are summed together to play a spectral sound.

> **NOTE**: because the Spectral Formant Filter depends on changing the balance among numerous discreet components, it can be used to filter a spectrally synthesised sound, but it cannot be used to filter the raw white noise signal produced by the Spectral Element in WHITENOISE mode. To create formant-filter effects in WHITENOISE mode, the best approach normally involves using the Source Filters in a parallel configuration (click for more details on this technique).

The **Spectral Formant Filter** can be applied to the **Spectral element** of any Source. To access it, click the Amp pop-up menu, scroll to 'Formant ModMap', and choose 'Fmt 1' (or create a new one). When you choose Fmt <N>, Alchemy will interpret the Nth ModMap as a description of how to boost/cut various frequencies: the horizontal (x) axis represents frequency, while the vertical (y) axis represents amplitude. For further details regarding the Spectral Formant Filter and the ModMaps that describe its behaviour, see the corresponding description of the Additive Formant Filter, which operates similarly.

13 Granular Resynthesis

13.1 A brief introduction to granular synthesis...

Granular synthesis represents continuous sound as a stream of 'grains', or tiny pieces of sound. The effects that result from this technique depend on a variety of factors: the duration and amplitude envelope of each grain, the degree of overlap or separation from one grain to the next, and of course the internal details of each piece of sound, such as pitch and pan.

Alchemy **generates grains** by extracting 2- to 230-millisecond pieces from an existing sample (e.g. a WAV or AIFF audio file). The amplitude of each grain is shaped by a simple **window**, and **modifications** may be applied to its pitch and pan before it is sent to the output stream.

If each new grain is extracted from slightly further into the sample than its predecessor, then the resulting stream of grains will essentially 'put' the pieces of sound back together' in a form that resembles the original source material. But a large number of interesting manipulations is also possible:

> **Time-stretching**. Grains can be sent out at a faster or slower rate than they are found in the original sample, causing it to play faster or slower without changing its pitch. You can even 'freeze' a sample at a certain point by extracting many grains from this one point.
>
> **Pitch-shifting**. Modifications to the pitch of each grain allow you to vary a sample's pitch without affecting its timing. By modulating the pitch and/or pan of each grain, you can also create effects such as spaciousness and blurring.
>
> You can **scramble** the order in which grains are played back to produce effects ranging from mild fuzziness to extreme mangling.

The Position and Stretch controls on the main Source sub-page permit a variety of time-stretching and -shuffling effects. (For details, see the manual's Source page, especially the example involving tempo-synced loops.) Other effects can be created by manipulating the Granular sub-page controls described below.

13.2 GRANULAR and SAMPLER mode buttons

The Granular element in each of Alchemy's four Sources provides two different types of sample playback, which you can select between using the **GRANULAR** and **SAMPLER** buttons on the left of the Granular sub-page. **Sampler mode** provides conventional sample playback, described in a bit more detail towards the bottom of this page. **Granular mode** provides granular sample playback and allows for a much wider range of manipulations using the Granular sup-page controls.

13.3 Granular sub-page controls

In addition to the controls listed below, granular-mode playback is also affected by different Loop modes and by the settings (and modulations) of the Position and Stretch knobs at the top of each Source sub-page.

> **Volume**. Adjusts the volume of the Granular element, independent of other elements in the same source. Unlike the remaining controls, which apply only in Granular mode, **Volume** applies in Sampler mode as well.
>
> **Size**. Adjusts the duration of each grain from 2 msec to 230 msec.
>
> **Density**. Controls the number of potentially overlapping grains from 1 (no overlap) to 10.

> **Note**: The key to using the Size and Density parameters effectively is to understand **how they interact**. When Density = 1, it's useful to imagine a single machine firing one grain at a time into the output stream. As soon as one grain finishes, the next one fires, so if Size = 100 msec, there's a new grain every 100 msec.
>
> Increasing Density to 2 adds a second 'machine' that fires its grains in between those of the first — so now there's a new grain every 50 msec, and the grains from machine 1 overlap those from machine 2. Further increases in Density cause new grains to fire more frequently and overlap more heavily.
>
> Setting Size around 100 msec and Density around 5 grains often works well for **smooth sounds without sharp transients, like pads**. Setting Size between 40 and 80 msec and Density around 2 grains often works well for **drums and other sounds featuring sharp transients**. Very small values of Size tend to produce a 'buzz' that masks the original pitch of the sample, while very large values of Size tend to have a 'rough and choppy' effect, but you can counteract both tendencies by increasing the Density. Finally, it takes more processing power to generate a larger number of overlapping grains, so you can **save CPU** by **decreasing** Density.

Window pop-up menu. Determines the envelope shape that is applied to each grain. Trial-and-error is really your best guide. Bear in mind that the effect of different Window choices is often quite subtle, especially when you avoid extremely high or low settings of Size and/or Density.

RTime. Applies a small random offset to the position in the sample from which each grain is extracted. (This is equivalent to modulating the Source Position knob with a RandGlide LFO and a low modulation Depth.) The default value of RTime is 3%, rather than 0%, because a small amount of randomisation helps to smooth the output of the Granular element.

RPan. Applies a random offset to the stereo position of each grain. (This is equivalent to modulating the Source Pan knob with a RandGlide LFO and a low modulation Depth.) You won't hear the effect of RPan unless the Source's Stereo button is engaged.

> **Note**: that **modulations applied to the Granular element** update with each new grain. This means for instance that modulating the Source Coarse Tune parameter with an LFO causes the stream of grains to rise and fall in pitch, but it does *not* create pitch sweeps *within* each grain. If granular Size is large and granular Density is low, modulations applied to Source parameters such as pitch may sound 'stepped' when you would expect them to sound smooth.

13.4 Using Sampler mode

Alchemy's Granular element can be switched to **SAMPLER** mode for conventional sample playback. In this mode, samples are played continuously rather than being split into grains. Transposing a sample upwards speeds it up, transposing it downwards slows it down. Sampler mode uses less CPU than granular mode, and imposes less colouration on the sound. **The trade-off**: granular manipulations of the sound are not available in sampler mode.

When Alchemy's Granular element is switched to sampler mode, the Source **Position** control determines the sample start point. You can modulate this parameter (e.g. with Velocity), and the modulation value at the moment a note begins will produce a corresponding offset in the sample start position. But changes in the modulation value are ignored as the note continues to play. The Source **Stretch** control likewise has no effect in sampler mode. Among the Granular sub-page controls, only the **Volume** control remains active in sampler mode; the other Granular sub-page controls work exclusively in granular mode.

14 Source Editor

The Source Edit page is accessed by clicking the **EDIT** button on any of the Source sub-pages (A, B, C, or D). It allows access to some of Alchemy's more specialised features.

On the left-hand side of the Source Edit page is a series of general controls.

The **A**, **B**, **C**, and **D** buttons select a source for detailed editing. The illuminated button corresponds to the currently selected source.

The **EDIT** button remains illuminated while the **Source Edit** page is in view. To **exit** this page and return to the current Source sub-page, click the **EDIT** button.

Several controls from the current Source sub-page are accessible here as well:

- The **ON** button toggles the source on (when lit) and off (when unlit).
- The Source Content selection field identifies the current source data (e.g. a file you have imported); click this field for access to a pop-up menu with a variety of options for working with Source content and settings. (See the Source page for a description of each of these options.)
- Additional controls let you adjust the **Loop** and **Keytrack** modes and toggle the **Stereo** setting. (Again, see the Source page for a description of each of these controls.)

The lower left-hand side of the Source Edit page lets you configure a Source with multiple Zones, each with a particular key range and velocity range. **Multiple zones** are created automatically whenever you load/import a multi-sample SFZ file.

The **Zone Name** selection field displays the file name of the currently active zone. The active zone is the one whose range and warp-marker details are currently displayed and editable. To view and edit range and warp-marker details for a different zone, click the zone selection field and choose the desired file from the pop-up menu.

Root Key determines which MIDI note will cause the zone sample data to play at its original pitch.

Low Key and **High Key** set the boundaries of the key range across which the active zone will play. For example, if you set High Key to E4, then the active zone will not play in response to MIDI notes from F4 upwards.

Low Vel and **High Vel** set the boundaries of the velocity range across which the active zone will play. For example, if you set Low Vel to 64 then the active zone will not play in response to MIDI notes with velocities less than 64.

> **Note**: when the key and velocity ranges of multiple zones overlap the first zone matching they current note will take precedence; it is not possible for multiple samples to be triggered simultaneously within a single source.

The **Show Markers** field determines which of the Warp Markers are superimposed on displays in the Main, Additive, and Spectral editors. You can choose to display 'All' markers, only the 'Loop' markers, or 'None'.

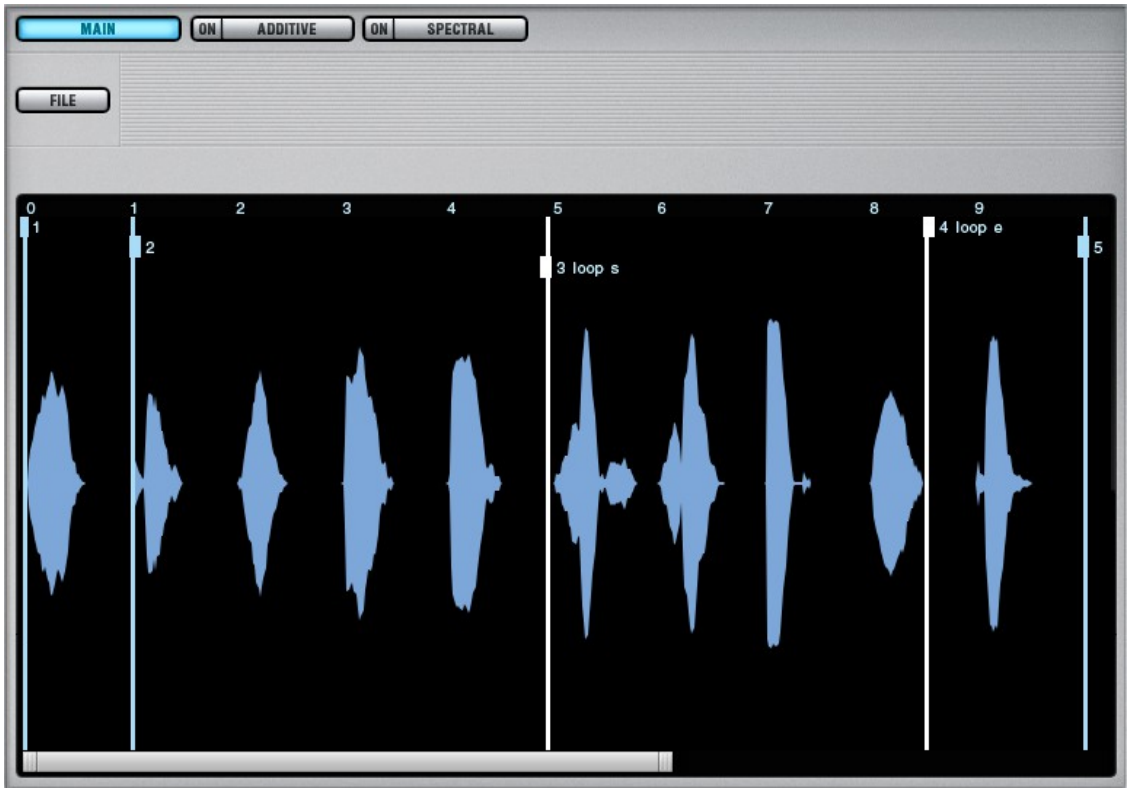
Choosing a value from the **Snap X** menu quantises warp marker adjustments to the selected number of evenly spaced divisions in the editor view.

The right-hand side of the Source Edit page contains a window for detailed graphical editing of Source data. Buttons across the top give you a choice between three edit modes:

MAIN - for working with the Source loop and warp markers. See below for details.

ADDITIVE - for editing additive data or creating it from scratch. See the Additive Editor page for details.

SPECTRAL - for editing spectral data or creating it from scratch. See the Spectral Editor page for details.



14.1 Using the Main Editor

Once audio data is loaded/imported into a Source, the Main Editor view provides an overview of the sound's natural amplitude envelope. A series of **warp markers** is superimposed on this overview. Each marker is depicted by a vertical line with a labelled handle that you can click and drag. The **Play Cursor** scrolls across the display, tracking the progress of the most recently played note.

> **Note**: it is not possible to adjust the position of the first or last warp marker (numbers 1 and 5 in the above image). **Double-click** the display to add new warp markers. The number associated with each existing warp marker, and the loop points, will be updated dependent on where the new warp marker was created.

14.1.1 What do warp markers do?

When you Morph from a sound with a fast attack to one with a slow attack, using **morph xy** or **time xy** mode, Alchemy **smoothly adjusts** the attack time according to the morph position. Warp markers define the boundaries of each sound's attack portion - and more generally, **warp markers define the boundaries of a series of time-aligned segments** when two or more sounds are morphed together. Furthermore, two of the warp markers define the **loop start** and **loop end** points of each sound.

Whenever you import additive, spectral, or granular data, Alchemy **automatically sets warp markers** at five positions: (1) the **very start** of the sound; (2) the end of the **attack** portion; (3) the **loop start** point; (4) the **loop end** point; (5) the **very end** of the sound.

> **Note**: that the effect of the loop start and end points depends on a Source's Loop mode. See the Source page for more information on Loop modes.

To **move** a warp marker, drag its handle. You can drag the loop start and loop end markers just like any other warp markers.

> **Note**: that you cannot position the loop start later than the loop end. However, it is possible to drag both the loop start and loop end markers to the same position, creating a 'sustain' point rather than an extended loop region.

To **insert** a new warp marker, double-click at the desired position.

To **delete** an existing warp marker, double-click its handle.

> **Note**: if you plan to experiment with the warp marker settings, it's a good idea to **Save** (or **Copy**) the Source, using commands from the pop-up menu in the source content field at the top left of the editor page (next to the ON switch). That way if you're not happy with the new settings you can restore the original ones by **Loading** the Saved source (or **Pasting** the Copied source).

The **FILE** button directly above the warp marker editor view offers commands to **Mark 8** = slices, **Mark 16** = slices, and **Mark 32** = slices. Each of these commands creates the designated number of warp markers and positions them at equal intervals across the entire sound.

14.1.2 Warp marker tips

One common reason for working with warp markers is to adjust the loop start and loop end markers. This can be useful for any sound that you want to play in a looped fashion. See the description of loop modes on the Source page for more details.

Tip: while the loop points may be adjusted using the warp marker editor, for **sample accurate** loop points e.g. to prevent clicks in **loop mode**, we recommend the use of an external audio editor capable of embedding loop points.

Adjusting the remaining warp markers has no effect on a single source. But if you are morphing between two or more sounds, these warp markers give you precise control over the time alignment of different portions of each sound.

- If your sounds are **single notes**, then the placement of the five default warp markers usually gives good results. But you may wish to experiment with the placement of marker 2, so that it's at the point where it sounds like the end of the attack for each sound. This can produce a more convincing morph between sounds with contrasting attack qualities.
- If your sounds consist of **multiple events** - e.g. musical phrases, drum loops, a spoken sentence - you may find it useful to create additional warp markers. If you are morphing between two voices speaking the same sentence, placing a warp marker at the start of each word will help to preserve the integrity of each word during the morph. If you are morphing between musical phrases or drum loops with different 'grooves' or timing nuances, placing a warp marker on every beat, half beat, or quarter beat, will help to ensure a smoother morph. (The 'Mark 8/16/32 = slices' commands give you useful starting points.)

15 Additive Editor

Once you have accessed the Source Edit page, clicking the **ADDITIVE** button at the top of the right-hand half of the page will bring Alchemy's Additive Editor into view.

15.1 Overview

The Additive Editor serves two purposes. First, it allows detailed editing of additive resynthesis data. And second, it allows you to design sounds from the ground up by creating the additive data yourself. In both cases, potentially large amounts of data are involved. Alchemy uses special graphical representations of the data in order to make these tasks manageable.

As explained on the main Additive page (see Additive), additive synthesis represents each sound as a sum of individual partials. The additive data describes each partial in terms of four parameters - amp, pitch, pan, and phase - each of which changes over time.

Therefore, you can think of the additive data as a series of 'snapshots', each of which captures the amp/pitch/pan/phase of every partial at a particular point in time.

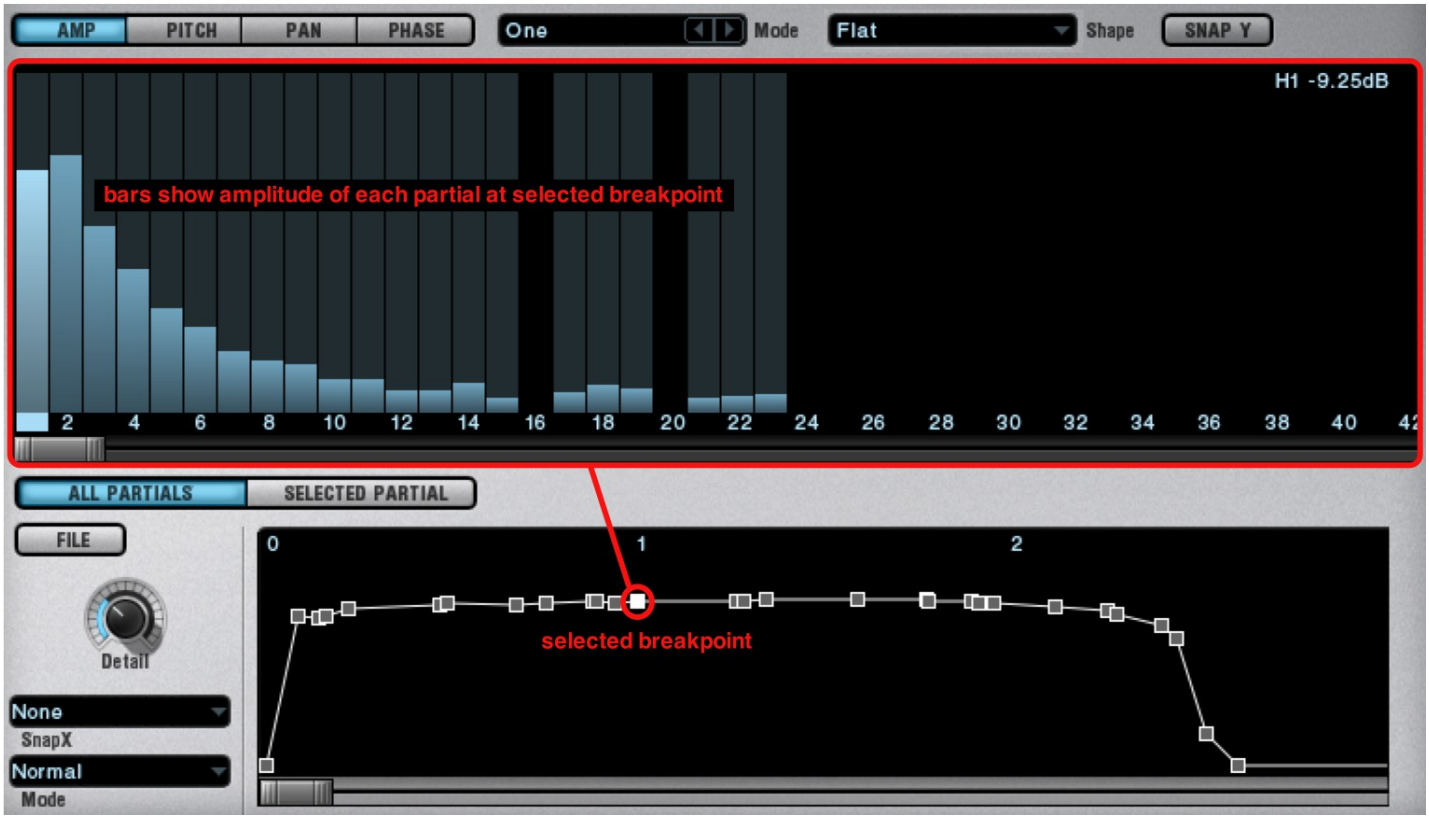
In between snapshots, each parameter updates smoothly towards its next snapshot value. When played in succession and with the right timing, the whole series of snapshots describes a potentially complex and continuously evolving sound.

> **Note**: Phase is not really an independent parameter. A partial's phase at any moment in a sound's evolution is determined by the phase at the beginning of the sound and by the (possibly changing) pitch of the partial. Since pitch information is captured in each snapshot, phase information doesn't have to be. The phase of each partial is only specified at the very beginning of the sound.

15.2 Breakpoints and partial bars

Alchemy presents the additive data using a pair of graphical displays. The first of these displays is a breakpoint envelope located in the bottom half of the Additive Editor view. Each breakpoint in this display corresponds to one of the 'snapshots' described above: it is positioned at a certain point in time, and it contains the amp, pitch, and pan values of every partial at that time-point. That, of course, is a lot more information than can be represented with a single point, which is where the second display comes into play.

The larger display, in the top half of the Additive Editor view, consists of a series of partial bars whose levels represent the amp, pitch, pan, or phase values of every partial, depending on the mode of the display. Normally, the values on display are those contained in the currently selected breakpoint - so the relationship between the breakpoint envelope and the partial bars display can be depicted as in the following image.



In the image above, the **partial bars** display shows the partial amplitude values contained in the currently selected breakpoint. By changing the mode of this display, you can also view the partial pitch or pan values contained in the currently selected breakpoint (as well as the initial phase value of each partial). Finally, the partial bars display can also be switched into an OVERALL mode, in which the partial amp/pitch/pan values are averaged across all the breakpoints. Working in OVERALL mode, you can quickly adjust the level of a certain partial across the entire sound, without having to step through individual breakpoints.

Back to the breakpoint envelopes. You now know that the timing (or x value) of a breakpoint represents the timing of a 'snapshot' that captures the amp/pitch/pan/phase values of every partial at that time-point. And you know that you can inspect all of the partial values contained in a breakpoint via the linked partial bars display. But what does the level (or y value) of a breakpoint signify?

Normally, when the partial bars display mode is AMP, PAN, or PHASE, breakpoint levels represent partial amplitudes. The precise meaning of a breakpoint level depends on the mode to which the breakpoint envelope itself is set. In ALL PARTIALS mode, each breakpoint level represents the sum of the partial amplitudes contained in a particular snapshot. If you raise or lower a breakpoint level in ALL PARTIALS mode, you'll see all the partial amp bars rise or fall accordingly. In SELECTED PARTIAL mode, each breakpoint level represents the amplitude of the one partial currently selected in the partial bars display. (The currently selected partial is indicated with a highlight in the partial bars display; note that partial number 1 is highlighted in the image above.) If you raise or lower a breakpoint level in SELECTED PARTIAL mode, you'll see only the currently highlighted partial amp bar will respond by rising or falling.

If, instead, the partial bars display mode is PITCH, then breakpoint levels represent partial pitches. Again the precise meaning of a breakpoint level depends on the mode to which the breakpoint envelope itself is set. In ALL PARTIALS mode, each breakpoint level represents the fundamental pitch. In SELECTED PARTIAL mode, each breakpoint represents the pitch of the currently selected partial.

> **Note**: If you look closely at the breakpoint envelope as you select different partials in **SELECTED PARTIAL** mode, you may notice (depending on the additive data) that the time positions of breakpoints vary from one partial to the next.
>
> **This is one of the keys to the high quality of resynthesis in Alchemy**: each partial can have an independent set of breakpoints. But this is not normally a concern for users. Whenever you work in ALL PARTIALS mode, Alchemy presents a series of breakpoints 'linked' to all of the partials, and adjustments to individual partial times and values are handled automatically behind-the-scenes.

15.3 Detail knob

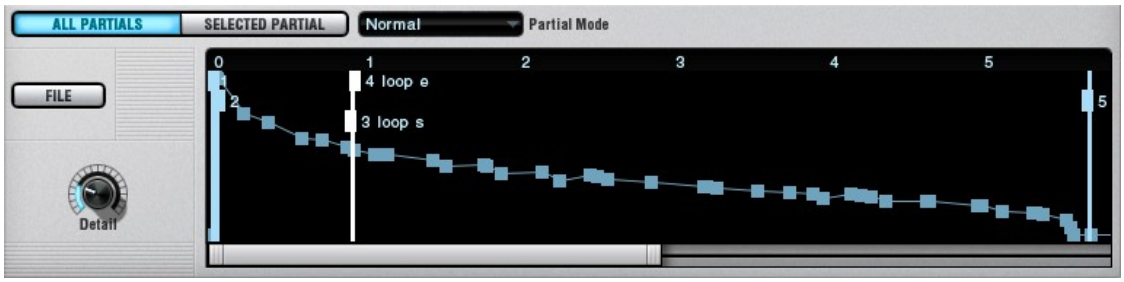
Let's import a factory sample in order to get a closer look at typical additive data. You can do this without exiting the Source Edit page, so we'll start from there.

If you are not already viewing the Source Edit page, access it by **clicking** the **Source A EDIT** button. Then click the ADDITIVE button across the top of the Edit page to bring the Additive Editor into view.

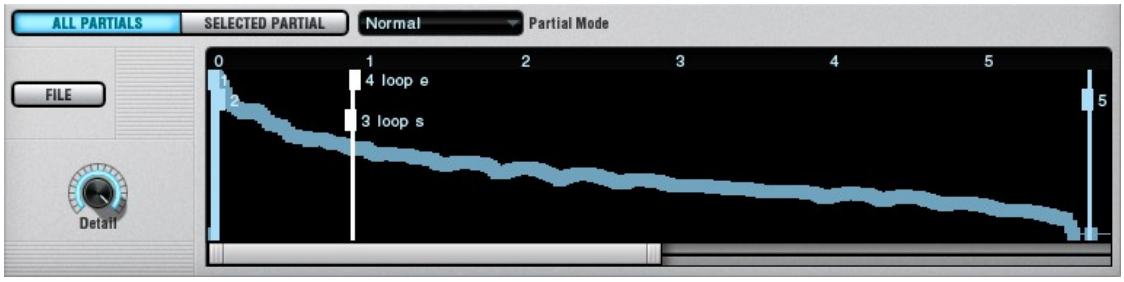
On the top left-hand side of this Source Edit page is a content field. (In an initialised preset, it reads 'Saw'; if you have already imported an audio file, it will show the filename; if you started with an initialised preset and have already made changes to the partial bars or breakpoint envelope, it will read 'EditorData'.) Click this field and choose **Import Audio** from the popup menu.

In the ensuing Import dialog, **set the import mode** to ADDITIVE and select FACTORY > Keys > Clavinet-FM > Clavinet-C3.wav. Leave the import settings at their default values and click IMPORT. (A more detailed description of this process is offered on the Import page.)

As soon as Alchemy has finished its calculations, you will be able to hear the resynthesised clavinet as you play your MIDI controller, and likewise you will be able to see the additive data displayed in the editor. We'll focus on the breakpoint envelope display at the bottom of the page. Set the **Detail** knob a value of 25%; the total number of visible breakpoints is reduced in the envelope - you may have to scroll or zoom the display to view them all.



In reality, there are many more data points at which partial amp, pitch, and pan values are defined. To see the complete series of breakpoints, increase **Detail** to 100% by turning the knob fully clockwise. You should now see something on the order of 500 breakpoints, but a precise count is beside the point. For anything other than the most excruciatingly precise sort of editing, the 100% view is more detail than you need to see.

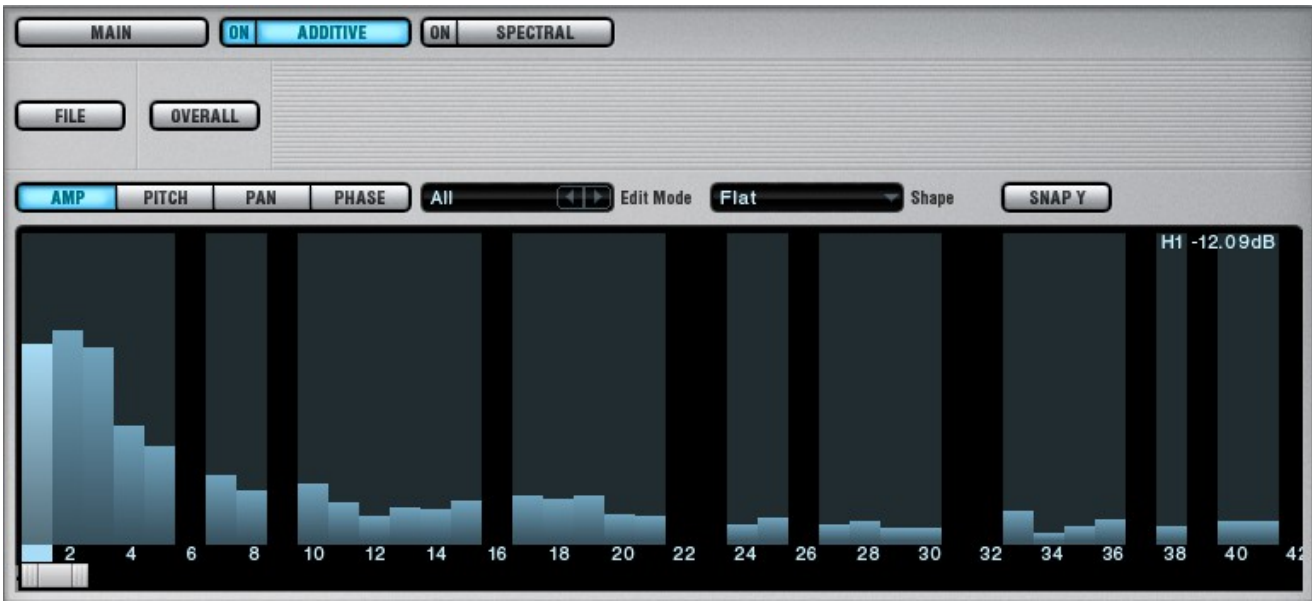


The beauty of the Detail function is that, no matter how high or low you set it, any adjustments you make — to a breakpoint time or level, or to individual partial amp/pitch/pan/phase values in the partial bars display — are chanelled through the currently selected breakpoint and applied appropriately to the underlying data.

> **Note**: If you are programming an additive sound from scratch, rather than editing resynthesis data, you may find it useful to work at **100% Detail**, so that you can see the exact data that you're creating.

15.4 Partial bars display and controls

Several controls are positioned above the partial bars display.



The **FILE** button offers various commands for transferring content from breakpoint to breakpoint:

Copy Breakpoints - Copies all the partial data contained in the currently selected breakpoint(s), and places it on the clipboard.

Paste Breakpoints - Pastes all the partial data from the clipboard to the currently selected breakpoint(s).

Paste Breakpoints Amp (**Pitch**, **Pan**) **Data** - Pastes one type of partial data from the clipboard to the currently selected breakpoint(s). If the partial bars display is set to AMP mode, the command is named 'Paste Breakpoints Amp Data', and so on.

Paste All Breakpoints Amp (**Pitch**, **Pan**) **Data** - Pastes one type of partial data from the clipboard to every breakpoint. If the partial bars display is set to AMP mode, the command is named 'Paste All Breakpoints Amp Data', and so on.

A common use of **Paste All Breakpoints Pitch Data** is in case a resynthesised sound has unwanted pitch fluctuations. First of all, you can resolve these artefacts by reducing the **PVar** (pitch variation) control on the Source A/B/C/D sub-page; but this aligns each partial with its harmonic tuning, which may change the character of the sound too much. So another approach is to use 'Copy from Breakpoint' to capture data from a single breakpoint, and then to 'Paste All Breakpoints Pitch Data'. Result: the partial pitch values of the copied breakpoint are applied to all breakpoints, so the inharmonic features of the copied breakpoint are preserved while the fluctuations from one breakpoint to the next are successfully eliminated.

When the **OVERALL** button is off (unlit), the partial bars display shows the values contained in the selected breakpoint. When it is on (lit), the partial bars display shows values averaged across all breakpoints.

The **AMP, PITCH, PAN,** and **PHASE** buttons determine which type of partial data is shown in the partial bars display.

The **Edit Mode** field contains options such as **One**, **All**, **Octaves**, and so on, which are described in more detail below.

The **Edit Shape** field contains the options **Flat**, **Bright**, and **Dark**, which are described in more detail below.

When the **SNAP Y** button is engaged, partial levels will snap to meaningful units when dragged. This is mainly useful when the display is set to **PITCH** mode, in which case partial levels will snap to semitones.

Several editing actions are performed directly in the display:

To adjust a **partial level**, drag it up or down, or click directly at the desired height of the partial bar.

You can **drag** left or right across multiple partials to set their levels with a single 'painting' action.

For **fine control** over a partial level, hold shift as you drag up or down.

The **top right corner** of the partial bars display reports the currently **selected partial number** and its pitch, amp, pan, or phase value.

To **set a partial level to zero**, you can drag it below the baseline; or you can right-click it.

You can **set multiple partial levels** to zero, once you've dragged a partial below the baseline, by keeping the mouse button pressed and dragging left or right. You can also 'zero' multiple partials by clicking and holding the right mouse button and then dragging left or right.

To **select a partial without changing its value**, click below the baseline. (Selecting a partial will bring its amplitude envelope into view when the breakpoint envelope display is in SELECTED PARTIAL mode.)

The **total number** of partials may be as high as 600. To work with partials that are currently out of view, use the standard scroll bar and/or zoom handles at the bottom of the partial bars display.

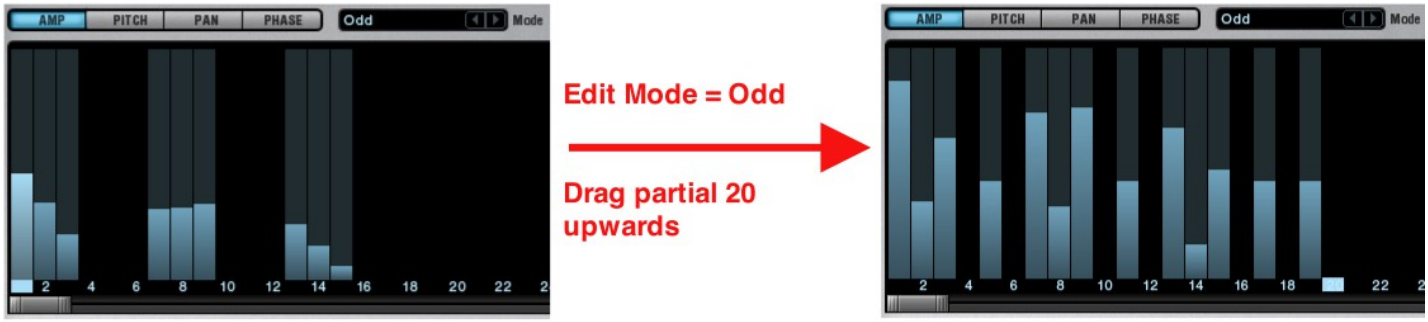
> **Note**: higher partials that are part of the additive data may not be heard unless the NOsc (number of oscillators) control on the Source A, B, C, D sub-page is set sufficiently high. For instance, raising the amplitude of partial number 72 will have no effect when NOsc = 60. Conversely, setting a higher NOsc value will have no effect in ADDITIVE mode unless there is actually data for the higher partials.

The **Edit Mode** field (above the partial bars display) lets you choose whether edits performed in the display apply to only one partial (in 'One' mode, which is the default) or to any of various groups (such as **Odd**, **Even**, and **Octaves**). For instance, if you choose the **Octaves** group and then drag upwards in the display, then the fundamental and all octave-related partials (2, 4, 8, etc.) will increase.

When **editing a group**, click/drag at the position of the highest partial that you want to be affected. For instance, if the **Edit Mode** is **Odd** and you drag partial 15 downwards, then the levels of the odd partials from 1 through 15 will be reduced.

When **editing a group**, hold the **control key** (Windows) or **command key** (Mac) to scale existing levels proportionally. When manipulated this way, **partials with higher levels will increase/decrease more**, **partials with lower levels will increase/decrease less**, and partials with zero levels will remain at zero.

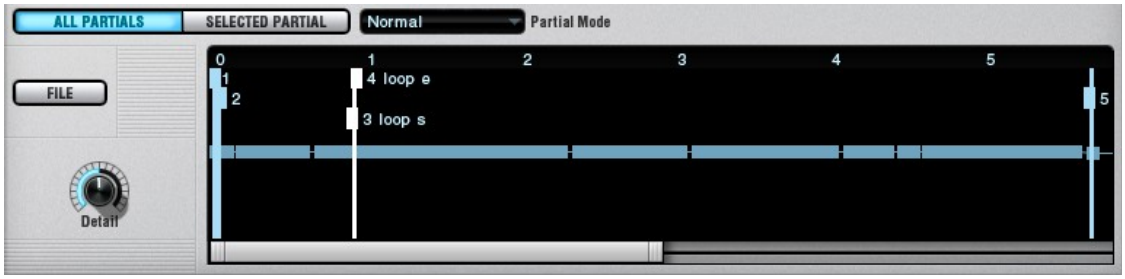
In contrast, **normal dragging adds to or subtracts from all partial bars equally**. The following pictures illustrate the difference between normal dragging and control-dragging (command-dragging).



The **Edit Shape** field (above the partial bars display) lets you choose whether edits applied to groups are applied evenly across the entire group (if Shape is set to 'Flat') or if they are weighted to favour the lowest partials of the group (**Dark**) or the highest ones (**Bright**).

15.5 Breakpoint envelope display and controls

The area above and to the left of the breakpoint envelope display is home to a number of controls.



The **FILE** button gives access to a pop-up menu in which you can **Clear all data** from the current Source, or translate a picture into additive data via the **Import Image** command, which is described in more detail later in this section.

A pair of **display mode buttons** determines how information is presented in the breakpoint envelope display.

In **ALL PARTIALS** mode, the breakpoint levels represent the sum of all the contributing partial amplitudes (or the rise and fall of the fundamental pitch).

In **SELECTED PARTIAL** mode, the breakpoint levels represent amplitude values of the one partial that is selected (and highlighted) in the partial bars display (or the pitch fluctuations of the currently selected partial).

The **Detail knob** controls the resolution of the breakpoint envelope display. It is described in more detail in a separate section.

The **Mode** field controls the behaviour of the envelope when a breakpoint is dragged.

Normal mode: other points remain stationary when one point is dragged.

Slide mode: dragging one point causes all the subsequent points the move in tandem, so that the relative distance between these points is preserved.

Stretch mode: dragging left compresses earlier points and stretches later points, while dragging right stretches earlier points and compresses later points; in either case the total length of the envelope is preserved.

Notice that breakpoint levels remain fixed in **Stretch** mode, so you can only drag left/right.

Working in the breakpoint envelope display is much like working with any of Alchemy's multi-segment envelopes (see the MSEG page for details). Here are some additional things you should know:

- The **time** and **level** of the selected breakpoint are reported in the parameter value display at the top of the interface.
- When the Additive Editor is in **PITCH** mode, breakpoint levels represent partial pitches. (You can edit these levels to eliminate a pitch wobble, or to create vibrato.) When the Additive Editor is in **any other mode**, breakpoint levels represent partial amplitudes.
- Unlike MSEGs and ModMaps, the additive breakpoint envelope always consists of **straight-line segments** with no convex/concave curvature.
- The very first and very last breakpoints always have an **amplitudes of zero**; in sounds with fast attacks, you'll find a second breakpoint with a nonzero amplitude at a time position very close to the first breakpoint. (You may need to increase the Detail setting in order to find that second breakpoint.)
- If you select **multiple breakpoints** (by dragging around them), then the partial bars display will display the **average** value of each partial across the selected breakpoints, and any edits you perform on the partials will be applied to all of the selected breakpoints.
- The **Loop markers**, or all of the warp markers, may be superimposed on the breakpoint envelope according to the setting of the Main editor's **Show Markers** field (at the far left of the Edit page, towards the bottom). Set this field to **None** if you want to hide all markers. For a full explanation of the warp markers, see the main Source Edit page.
- When you **drag warp markers** in the breakpoint envelope display, they always snap to existing breakpoints. (Compare the main Source Edit page, where warp markers can be dragged freely to any position.)

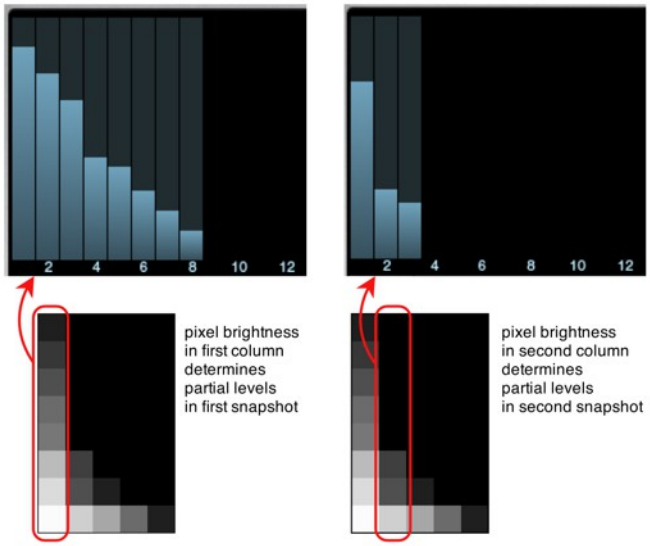
- Positioning the **loop start** and **loop end** markers both on the same breakpoint creates a sustain point, which means you can listen to that point in the sound for as long as you wish. By doing this repeatedly at different breakpoints, you can 'investigate' successive snapshots of the additive data in order to locate ones you may wish to edit.

15.6 Image import

Alchemy is capable of importing images (in PNG, or Portable Network Graphics, format) and translating them into additive synthesis data. You can easily convert existing images into sounds, and you can design new sounds by using (external) graphics software and drawing the image you want to import.

Note that Alchemy's Spectral Editor has its own image import capabilities, and you can get characteristically different results by importing an image into either the additive or the spectral engine.

The translation from image to additive data works as follows:



- Each **column of pixels** represents a **snapshot** in the additive data. The leftmost column describes snapshot 1, the next column to the right describes snapshot 2, and so on. (These snapshots are timed at a steady rate of 20 per second.)
- Within **each column**, there is a one-to-one correspondence between **pixels** and **partials**: the bottom row represents partial 1, the second row up represents partial 2, and so on. Thus the height of your image determines the number of partials — an image **100 pixels tall**, for instance, translates to a sound using **100 partials**.
- The **brightness** of each pixel determines the amplitude of a particular partial in a particular snapshot. A **black pixel** corresponds to **silence**, while a **white pixel** corresponds to **maximum amplitude**. (You can import both colour and grayscale images; colour information other than brightness is discarded.)

Importing an image with many bright pixels gives you additive data with many high-amplitude partials, which may cause clipping to occur. If the results of an image import are unexpectedly noisy, you can reduce the overall amplitude in a number of ways:

- **Reduce** the Vol setting of the additive element.
- **Edit** the data by setting the Additive Editor to **OVERALL** mode, selecting the **All** edit mode, and dragging downwards in the partial bars display.
- Use your **graphics software** to darken the image before importing it.

15.7 Example: Creating a simple additive sound from scratch

In this example, we'll introduce some basic principles of additive programming in Alchemy by creating a simple **plucked** sound from scratch.

Initialise Alchemy by choosing the **Clear** command in the Title Bar **FILE** menu.

Go to the **Source A** sub-page, just to get a better look at the current state of the preset.

When initialised, Source A is in VA mode, the oscillator waveform is set to the basic **Saw**, NOsc = 1, and the Pitch profile is set to **Unison**.

For additive synthesis, we normally want an entirely different configuration:

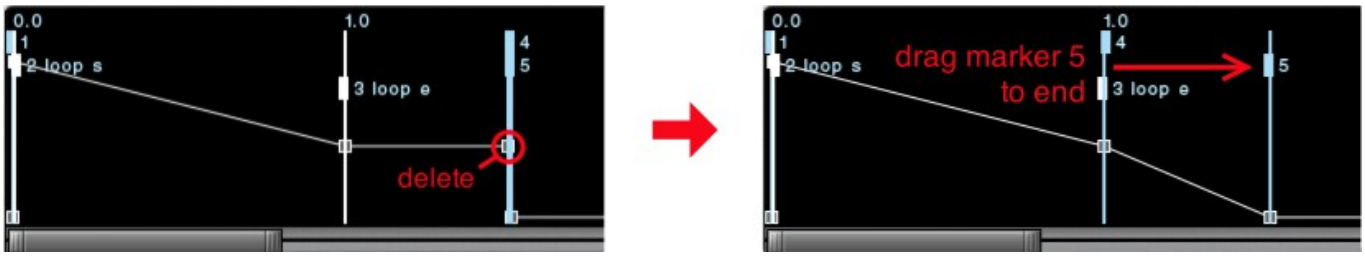
- We need Source A to be in **ADD** mode.
- We typically want the oscillator waveform set to the basic **Sine**, with a larger value of NOsc (permitting many partials to be active).
- The Pitch profile is set to **Harmonic**.

> **Note**: it is not necessary to change any of these settings from their initial values - as soon as you enter the Additive Editor and start creating data, the configuration of Source A switches, automatically, to a sensible set of defaults for additive programming. So let's get right to work…

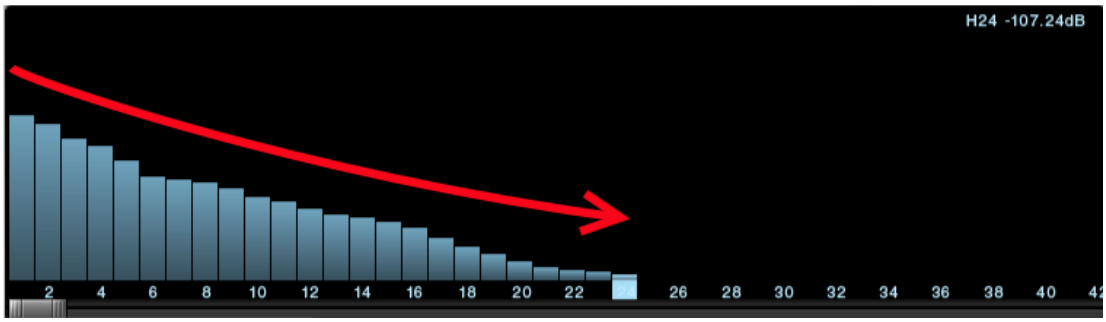
Click the Source A **EDIT** button to open the **Source Edit** page, and then click **ADDITIVE** at the top of the page to bring the Additive Editor into view.

Since we're starting from scratch rather than editing resynthesised data, let's increase **Detail** to **100%** for an accurate view of all the data. At this point the breakpoint envelope display won't appear any different, since there are not yet any additional details to be shown.

If the **second breakpoint** in the envelope is not already selected, **click it**. This is a typical starting point, since the first breakpoint always sits at time zero and level zero, and can't be altered.



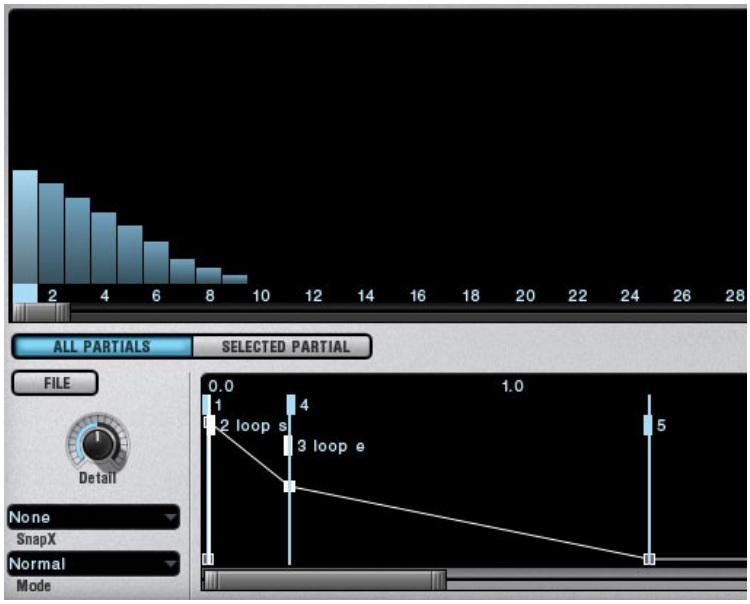
Now ensure that the **partial bars** display is set to **AMP** mode, and that **OVERALL** mode is **off**. Draw in some bars to define the harmonic content of the beginning of the sound. Since our goal is a **plucked** sound, we want a **bright sound** with a significant number of **active partials** at the **beginning**, which will then **decay** to a **darker** sound further on. A design such as the following one - which can be created in a single sweep across the display - works well, but you needn't copy it precisely.



Play a few notes on your MIDI controller to confirm that the sound now **begins** with a **bright timbre** as intended. This is a step in the right direction, but more needs to be done. For instance, the default loop mode is **Continuous**, so the sound will loop indefinitely when you hold a note. To switch looping off, change the **Loop mode** to **None** in the field at the far left of the Edit page.

Progress! But now the end of the sound falls off too abruptly instead of decaying smoothly. A quick solution is to delete the second-to-last point. Afterwards, drag warp marker 5 back to the final point.

Now the results sound reasonably good. But let's **re-shape the decay** so that most of the energy falls off quickly and the **tail** of the decay is less dramatic. Select breakpoint 2 again, and then choose **Copy Breakpoint** from the partial bars display **FILE** pop-up menu. Then select breakpoint 3, drag it most of the way towards breakpoint 2, and choose **Paste Breakpoint** from the partial bars display **FILE** pop-up menu. Then reduce or eliminate the higher partials in the data you've just pasted. With breakpoint 3 still selected, the final result should look something like the image to the right.



16 Spectral Editor

Once you have accessed the Source Edit page, clicking the **SPECTRAL** button at the top of the right-hand half of the page will bring Alchemy's Spectral Editor into view.

16.1 Overview

The Spectral Editor serves two purposes. First, it allows graphical editing of spectral resynthesis data. And second, it allows you to design sounds from the ground up by working with simple **paint** tools directly on the **canvas** of the graphical display.

Spectral data is displayed in the editor according to the following principles:

Time (in sec) is represented along the *x-axis* (from left to right).

Frequency (in Hz) is represented along the *y-axis* (from bottom to top).

Amplitude is represented by brightness, using shades of blue. The full range, from silent (black) through loudest (white) looks like this:

> **Note**: accurate resynthesis requires a much finer frequency resolution than the spectral display can accommodate. Therefore the frequency information depicted in the display is a somewhat coarse representation of the underlying data. Editing and creating data graphically is performed at the resolution of the display. This means, for instance, that you cannot paint conventional melodies and chords consisting of precise notes.

16.2 Spectral editor controls

Click the **FILE** button to access a pop-up menu with the following commands:

Import Image — Opens a dialog in which you can select a file in PNG format to be placed as an image on the spectral canvas. The imported image is placed at the far left of the canvas. The height of the image is scaled to fit the entire vertical range of the canvas, and the width of the image is scaled by the same factor as the height (so the proportions of the original image remain intact). The imported image's colour information is discarded, and its brightness information is mapped to the shades-of-blue amplitude scale described above. If the newly placed image does not extend fully to the right-hand edge of the canvas, then existing data that lies beyond the right edge of the newly placed image remains in place. Images with a height of 256 lines will result in a 1:1 mapping of pixels to spectral bins. The spectral editor actually displays the image as 381 lines, so when editing within Alchemy there will be some blurring between bins.

Import Image to Brush — Opens a dialog in which can select a file in PNG format to be used as a brush for painting on Alchemy's spectral canvas. Unlike importing an image directly onto the canvas, importing it as a brush preserves the original dimensions of the image and gives you full control over the position at which the image is placed. (As with any brush, you can 'print' an image of the imported brush by clicking or paint strokes with it by dragging.)

Undo — Reverses the last change made to the spectral data.

Clear — Deletes all spectral data from the source, leaving only silence (solid black image).

SELECT and **BRUSH** buttons — When the editor is in SELECT mode, clicking and dragging across the spectral canvas defines a rectangular selection. If you then switch to BRUSH mode, the selection you've just defined can immediately be used as a brush shape, which remains available until you select a different brush shape.

The **Shape** field offers a choice of several pre-defined brush shapes, each with a descriptive name.

The **Scale** field offers a choice between **Linear** and **Log2** scaling for the spectral display. **Log2** mode is useful for increasing the size of the low-frequency spectrum display for editing purposes.

The **Colour** knob adjusts the amplitude scaling of the brush. White brush pixels paint white on the canvas when Colour is 100%, medium blue on the canvas when Colour is 50%, and black on the canvas when Colour is 0%. Brush pixels representing lower amplitude values are scaled accordingly; for example, medium blue brush pixels paint medium blue on the canvas when Colour is 100%, dark blue on the canvas when Colour is 50%, and black on the canvas when Colour is 0%.

The **Opacity** knob allows for transparency effects. When Opacity is 100%, the paint modes behave as described below (see 'Paint mode buttons'). When Opacity is 50%, the pixel values that result from painting in any mode are halfway between the pixel values in the existing image and the pixel values that would result from painting with full Opacity. (Note that an image imported directly to the canvas will always have full Opacity; on the other hand an image imported as a brush can be used to paint with any degree of Opacity.)

The **Paint mode** buttons determine how the brush interacts with the existing canvas image.

In **SET** mode, each pixel on the canvas is set identically to the corresponding brush pixel. In this mode, painting with black replaces the existing image with silence.

In **ADD** mode, the brush pixel values are added to the existing canvas pixel values. In this mode, painting with black leaves the existing image unchanged, while painting 'multiple coats' of blue results in higher and higher amplitudes the more coats you paint.

In **MUL** mode, the brush pixel values are multiplied with the existing canvas pixel values. In this mode, painting *with* black on an image of any colour, and painting *on* black with paint of any colour, both result in black. You can use a black brush in MUL mode as an eraser, and you can use a brightly-coloured brush in **MUL** mode to boost the portions of an image that already produce sound, without disturbing the portions that are silent.

Finally, in **DEL** mode, the brush pixel values are subtracted from the existing canvas pixel values. In this mode, a bright white brush works like an eraser.

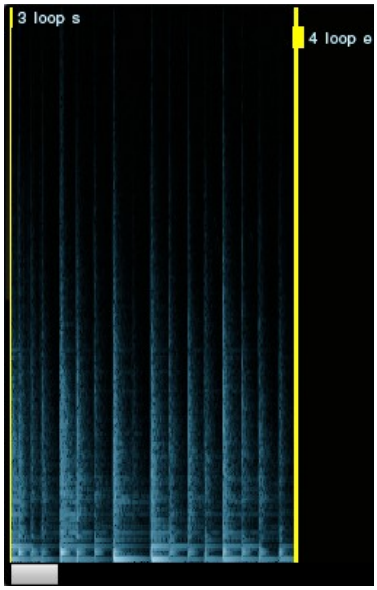
16.3 Spectral canvas

Once you are familiar with the brush modes, the relation between colour and amplitude, and the distribution of frequencies from the bottom to the top of the canvas, working in the spectral editor's graphical display is an intuitive process. Here are some useful techniques and related information.

- The **time** (in sec) and the **frequency** (in Hz) of the brush (or selection cursor) position are reported in Alchemy's parameter value display. If data is present at, say, the 2500 Hz position, this means that the output of the spectral element will include a 2500 Hz component when C3 is played (or when key-tracking is off and any note is played).
- To constrain painting **vertically**, hold **Shift** while painting.
- To constrain painting **horizontally**, hold **Control** (Windows) or **Command** (Mac) while painting.
- **Warp markers** are superimposed on the canvas and can be moved freely by dragging their handles. To create or delete warp markers, or re-assign the loop start and end markers, switch to the main Source edit view by clicking the MAIN button at the top of the editor.

A **Play Cursor** scrolls across the display, tracking the progress of the most recently played note.

16.4 Example: Modifying a drum loop



Initialise Alchemy by choosing the **Clear** command in the Title Bar **FILE** menu.

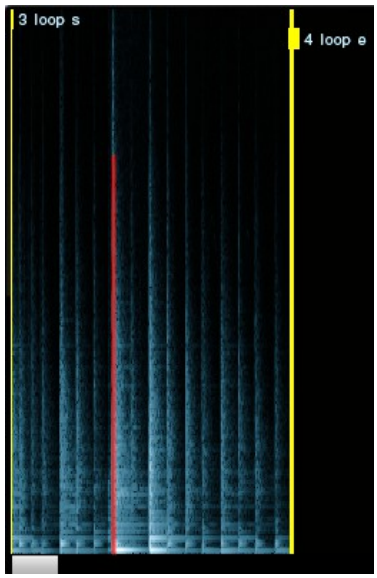
Import (into Source A) the factory sample Loops > PercussiveLoops > DjembeLoop2-4bts. Be sure to set the import mode to **SPECTRAL**. (See the Import page for more details of this process.)

The imported image should look similar to the first image on the right.

- **Play** and hold note C3 to get a feel for what the loop sounds like without any changes.

Select the brush shape **Transient**, which is designed for creating or boosting attack transients. We'll use this tool to add 'punch' to the four drum hits marked above.

- Set the **Colour** knob turned to approximately 75%.
- Reduce the **Opacity** to around 25%.
- Set the brush mode to **MUL**.



Carefully align the brush so that its **bright left edge** coincides with the **peak** marked in the image to the right.

Click one or more times, until the result looks something similar to the example.

Each click of the mouse will **multiply** the area under the **Transient** brush, thereby increasing the amplitude of the spectral data. Because we are using **MUL** (multiply) mode, areas with an amplitude of zero will **not** be affected (because there is nothing to multiply).

Play and hold C3 on your MIDI controller to hear the effect of these edits.

Assuming the example has been followed correctly, the middle of the loop should now be emphasised significantly, creating a new rhythm!

Let's make some additional changes to the end of the loop:

With the **Transient** brush still selected, and the brush mode still set to **MUL**, reduce the Colour value all the way to black (Colour knob turned fully left).

Align the brush with the position one sixteenth note **after** the final beat; click at this position, and drag to the right to erase all the spectral data from this position to the end.

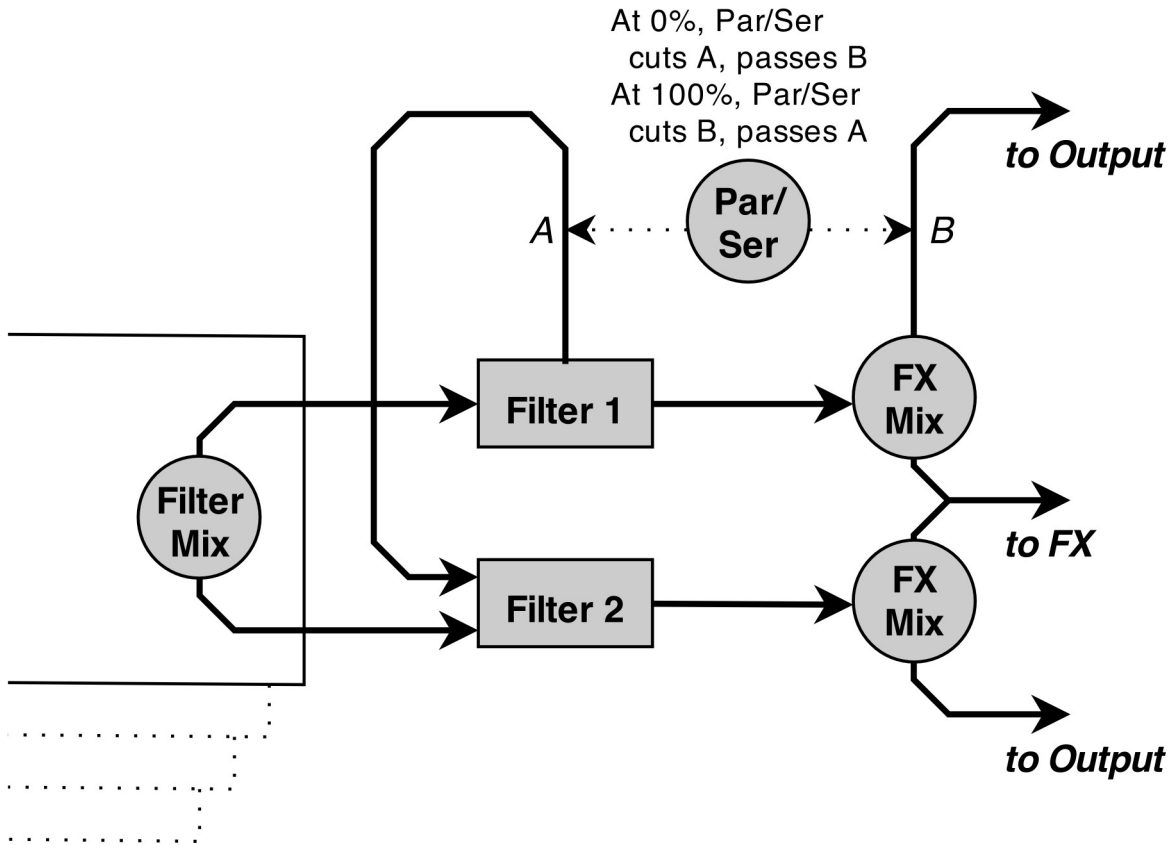
- **Play** and hold C3 again on your MIDI controller to confirm that this produces a silent region at the end of the loop.

Switch to the **SELECT** tool and select the first half of a medium-loud drum hit. Switch back to the brush tool, turn the Colour knob fully right, and set the brush mode to SET. You can now paint using the selected image as a brush. Click four adjacent positions at the end of the loop to produce a ''roll' effect.

- **Play** and hold C3 once more on your MIDI controller to hear the final result.)

17 Main Filters

There are two Main Filter modules, labelled **Filter 1** and **Filter 2**.

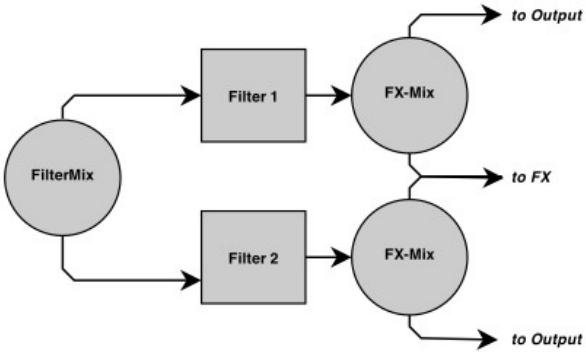


The outputs from all four Sources are sent (in varying amounts, according to their respective FilterMix settings) to the Main filters. Both Main filter modules provide multi-mode filters with identical controls. Each one has its own **FX-Mix** control, which (like the FilterMix control in a Source) works as a kind of two-channel mixer, or crossfader.

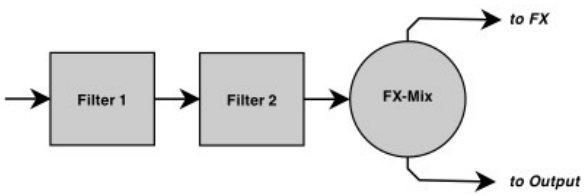
- FX-Mix control turned **all the way to the left**: the filter's output is sent to Alchemy's main output, and none of it to the Effects module.
- FX-Mix control turned **all the way to the right**: all of the filter's output is sent to the Effects module, and none of it to Alchemy's main output.
- FX-Mix control **set half-way**: equal amounts of signal are sent to the Effects section and Alchemy's main output.

Finally, the **Par/Ser** knob acts as one more crossfader, allowing you to send the **non-FX** portion of the Filter 1 signal to the Output stage (when Par/Ser = 0%), to the input of Filter 2 (when Par/Ser = 100%), or to a mix of these destinations (when Par/Ser has some intermediate value). Therefore:

To use the Main filters in parallel - set the Source FilterMix controls as desired, set Par/Ser to 0%, and set the Filter 1 FX-Mix control as desired.



To use the Main filters in series - set the Source FilterMix controls to '100% F1 0% F2', set the Par/Ser knob to 100%, and set the Filter 1 FX-Mix control to 0%.



> **Note**: that when the Par/Ser knob is set to 100%, a portion of the Filter 1 signal will bypass Filter 2 whenever the Filter 1 FX-Mix control is set above 0%.

17.1 Global page controls

The Filter section on the Global Page provides access to the two Main filter modules, each of which offers a powerful multi-mode filter.

Identical controls are provided for Filter 1 and Filter 2:

On toggles the filter unit on or off.

The **filter type** selection field allows you to set each filter to one of 50 filter types (see 'Filter Types' below).

The three ***primary filter controls*** are **Cutoff**, **Res**, and **Drive**. The meaning of these controls depends on the selected filter type (see 'Filter Types' below).

FX-Mix controls how much of the filter module's output is sent to Alchemy's Effects section, and how much directly to the main Output.

17.2 Filter VU meters

Both Main Filters include VU meters indicating the current audio level received from all four sources combined.

When the output from **all four sources** exceeds 0dB the VU meters will indicate clipping has occurred by momentarily turning red.

Clipping is an undesirable artefact of digital audio. Clipping occurs when the output exceeds 0dB or the input to an effect exceeds the maximum range, producing a nasty distortion like sound in the output stage. If clipping does occur, simply reduce the amplitude of the loudest source or adjust the balance of the sources using the Morph section.

17.3 Filters and filter types

Altogether, Alchemy provides a total of 15 multi-mode filter modules:

- Two **Main Filters**
- Four sets of three **Source Filters**
- A **MMFilter** module in the Effects section

There are 50 filter types to choose from in the Main Filters and the MMFilter effects module. The Source Filters offer a smaller selection of filter types, but they lack the Drive control found in the Main Filters and the MMFilter module.

You will find descriptions of all 50 filter types below.

17.3.1 Filter types: lowpass, bandpass, highpass

These filter types will be familiar to many users. A **lowpass (LP)** filter passes the portion of a sound below a specified cutoff frequency and rolls off the portion above that frequency. A **bandpass (BP)** filter passes the portion of a sound occupying a band surrounding the cutoff frequency and rolls off the portions above and below that band. And a **highpass (HP)** filter passes the portion of a sound above a specified cutoff frequency and rolls off the portion below that frequency.

There are 42 different LP, BP, and HP filter designs available in Alchemy, each with distinctive characteristics that you may prefer for a given purpose. The three **principal filter controls** have standard functions for all 42 of these filters:

Cutoff controls the filter cutoff frequency.

Resonance controls the filter resonance or emphasis, such that higher settings boost frequencies in the immediate vicinity of the cutoff frequency.

Drive allows the filter to be overdriven; the precise effect varies with each filter design.

The available LP, BP, and HP filter designs include:

LP2-BQ, **BP2-BQ**, **HP2-BQ** - 2-pole, bi-quad filters.

LP2-Fat, **BP2-Fat**, **HP2-Fat** - 2-pole filters designed to saturate heavily at higher Resonance and Drive settings.

LP2-SVF, **BP2-SVF**, **HP2-SVF** — 2-pole state-variable filters.

Analog-modelled filters — there are several designs modelled on the filters in classic hardware synths. Each of these is available in **two versions**, a normal one and a 2x-oversampled **HQ** one which offers higher quality at the cost of greater CPU usage.

2-pole analog-modelled filters — **LP2-K20**, **BP2-K20**, **HP2-K20**; **LP2-XP**, **BP2-XP**, **HP2-XP**; **LP2-MG**, **BP2-MG**, **HP2-MG**.

4-pole analog-modelled filters (these provide a steeper roll-off of frequencies beyond the cutoff) — **LP4-XP**, **BP4-XP**, **HP4-XP**; **LP4-MG**, **BP4-MG**, **HP4-MG**.

17.3.2 Filter types: formant and peaking

The **formant filter** works like a bandpass filter with a potentially very narrow bandwidth. It is designed to mimic a formant, or fixed resonance, such as those produced by the hollow body of a guitar or by the shape of the vocal cavity (which is adjusted during speech to produce different vowel sounds).

Cutoff controls the resonant frequency of the formant.

Resonance controls the width of the band surrounding the resonant frequency. Low values produce a narrow band, high values a wider one. Setting the resonance too low may cause little or no sound to pass through the filter.

Drive has no effect for this filter type.

> **Note**: To create multiple formants, you can configure the Source Filters in parallel and set the type of all three filters to **Formant**. See the example at the bottom of this page, which uses multiple formants to mimic vowel sounds.

The **peaking filter** boosts a narrow band around some resonant frequency while leaving the rest of the signal more or less unchanged.

Cutoff controls the resonant frequency of the boosted band.

Resonance controls the amount of boost. Higher values are generally the most effective.

Drive has little effect for this filter type beyond boosting the overall gain.

17.3.3 Filter types: comb filters

A **comb filter** mixes the original signal with one or more copies delayed by a very short time interval. At some frequencies this mixture causes phase cancellations, while at other frequencies it causes reinforcements; the result is a 'spiky' frequency response with multiple resonant peaks. (A graph of these peaks resembles the teeth of a comb, which gives this filter type its name.)

Alchemy offers two comb filter designs. **CombP** (P is for 'plus') uses positive feedback on the delay lines, while **CombM** (M is for 'minus') uses negative feedback to produce less extreme effects, often with a 'hollow' quality.

Cutoff controls the delay time in the comb circuit (lower cutoff = longer delay).

Resonance controls the amount of feedback in the comb circuit.

Drive has little effect for the comb filter types beyond boosting the overall gain.

> **Note**: sending a percussive sound into a highly resonant comb filter will cause it to **ring** at a frequency determined by the delay time (hence controlled by the Cutoff knob).

17.3.4 Filter types: ring modulation

Ring modulation is a process in which a modulator and a carrier signal are multiplied. Each frequency component of the modulator interacts with each frequency component of the carrier to produce two sidebands: a sum and a difference (carrier – modulator). When an Alchemy filter is set to the **RingMod** type, the signal entering the filter acts as the modulator, while the carrier is supplied internally by the filter.

Cutoff controls the carrier frequency.

Resonance applies a constant offset to the carrier.

- At **0% Resonance**, the carrier wave varies between –1 and +1, and the result is classic **ring modulation**.
- At **100% Resonance**, the carrier wave varies between 0 and 1, and the result is classic **amplitude modulation**. In this case, the carrier signal itself is present alongside the sum and difference sidebands.

Drive controls the carrier waveform.

- **Setting Drive to 0%** gives a ***pure sine wave*** carrier. This setting is often the most useful for producing characteristic 'bell-like' timbres.
- **Setting Drive above 0%** progressively ***truncates*** the carrier waveform (reducing it to just the first quarter of the sine shape when Drive is 100%). These settings produce a greater density of sidebands, typically resulting in noisy/grungy timbres.

17.3.5 Filter types: distortion

The Alchemy filters offer several distortion effects.

Tube — implements the well-known and well-loved 'Tube' distortion effect from CamelPhat.

Mech — A more 'metallic' sounding distortion, also from CamelPhat.

BitRed — A harsher sounding 'bit reduction' effect (also known as 'bit crushing').

> **Note**: distortion effects created in the Source Filter and Main Filter stages of the signal path are ‘polyphonic’ — that is, each voice is distorted independently, so there are no intermodulation effects when you play chords with this kind of distortion. In contrast, the Distortion module in the Global Effects stage at the end of the signal path processes a mix of all the voices.

The principal filter controls work as follows when the filter type is set to a distortion effect:

Cutoff controls the intensity of the distortion effect.

Resonance controls the mix between clean and distorted signals (0% = clean only, 50% = equal mix, 100% = distorted only).

Drive has little effect for the distortion types beyond boosting the overall gain.

17.4 Example: Creating vowel sounds with parallel source filters

Vowel sounds in human speech are the result of formants produced by changing the shape of the vocal cavity. By mimicking these formants, Alchemy can produce speech- or song-like effects, even in VA mode (i.e. without relying on samples of actual speech). Here is a basic example.

Initialise Alchemy by choosing the **Clear** command in the Title Bar **FILE** menu.

Go to the Source A sub-page and adjust the following settings:

Increase **Amp** to 0 dB.

Set **Coarse Tune** to –12 semis.

Click the Source A **Fine Tune** knob to bring its mod rack into view in Alchemy’s Mod section.

In the first mod-rack slot, choose ‘LFO’ > ‘LFO 1’, and adjust the mod depth to approximately 50%.

In the LFO 1 control panel, set **Attack** to approximately 0.50 sec, turn off **SYNC**, and set **Rate** to approximately 5 Hz.

Turn on the ***Source A Filter section,*** and click the Filter button to bring the Source Filter details into view.

Turn on all three of the ***individual Source Filters***, and set the configuration to **parallel**. We’ll come back to these filters and adjust the Cutoff and Resonance settings in a later step.

Switch the Source section view to ALL. Click in the ***Source A content field*** and choose 'Copy Source'. Then click in the ***Source B content field*** and choose 'Paste Source', click in the ***Source C content field*** and choose 'Paste Source' again, and click in the ***Source D content field*** and choose 'Paste Source' a final time.

On each of the Source Filter sub-pages, adjust the following settings (all values are approximate):

Source A — 'Ah' as in *father*

- Cutoff 1 = 800 Hz, Res 1 = 100%
- Cutoff 2 = 1200 Hz, Res 2 = 100%
- Cutoff 3 = 2800 Hz, Res 3 = 100%

Source B — 'Ee' as in *peace*

- Cutoff 1 = 230 Hz, Res 1 = 100%
- Cutoff 2 = 2600 Hz, Res 2 = 75%
- Cutoff 3 = 3200 Hz, Res 3 = 75%

Source C — 'Oo' as in *food*

- Cutoff 1 = 200 Hz, Res 1 = 100%
- Cutoff 2 = 880 Hz, Res 2 = 65%
- Cutoff 3 = 2400 Hz, Res 3 = 50%

Source D — 'Eh' (as in *let*)

- Cutoff 1 = 530 Hz, Res 1 = 65%
- Cutoff 2 = 1850Hz, Res 2 = 75%
- Cutoff 3 = 2500Hz, Res 3 = 50%

Now that each Source is configured to produce a vowel sound, let's set up **morphing** between these sounds and control it via the modwheel.

Switch the Source section view once more to **ALL**. Set the **MORPH** mode to **morph linear** and set the X knob (the only knob) to 0%.

The **Morph X** knob's mod rack should now be in view in the Mod section. Since the modwheel is linked by default to Performance Control 7, click in the first mod-rack slot and choose Perform > Control7. Leave the mod depth at its default value of 100%.

Now as you play notes, you can use the **modwheel** to morph between vowel sounds.

If the output level seems low, you can **boost** it by increasing the Master Amp and/or Volume knobs. (And a further boost, should you need one, is available if you load a Compressor module in the effects rack and adjust the Amount knob.)

18 Master Section

Once the signal path of an individual voice in Alchemy has left the Source section and passed through the Main Filters, it reaches the **Master** section, where controls are provided for the Amplitude, Pan, and Coarse and Fine tuning of the whole voice before it mixes with other voices at the input of the Effects stage.

The Master section also provides high-level control over various attributes of a preset such as polyphony and micro-tuning. The available controls are described below.

18.1 Master voice controls

Amp — Adjusts the level of the voice. By default, AHDSR 1 modulates this parameter, thereby acting as the master amplitude envelope.

> **Note**: Whenever an AHDSR is assigned to modulate Master Amp, the modulation depth is 'locked' and can't be adjusted. This is because setting the depth below its default of 100% would cause notes to remain above zero amplitude indefinitely — so it protects against 'stuck' notes. (If you replace the AHDSR with another type of modulator, such as an MSEG, you'll want to preserve a mod depth setting of 100% unless you really intend for notes to sustain indefinitely.)

Pan — Adjusts the stereo position of the voice. (Acts as a 'pan' control for mono sounds and as a 'balance' control for stereo sounds.)

Coarse and **Fine Tune** — Adjusts the pitch of the voice in semitones (Coarse) and cents (Fine).

18.2 Other Master controls

Trigger Mode field - determines the conditions under which (1) a trigger signal is generated and (2) a portamento glide occurs. (Interacts with the Voice and Porto settings, described below.)

Always. ***If Voices = 1***, then a trigger is generated at the start of each legato group, and portamento occurs at the start of every note. ***If Voices > 1***, then a trigger is generated at the start of every note, and portamento occurs at the start of every note.

Retrigger. Regardless of the Voices setting, a trigger is generated at the start of every note, and portamento occurs at the start of every note.

Legato. ***If Voices = 1***, then a trigger is generated at the start of each legato group, and portamento occurs at the start of each legato group. ***If Voices > 1***, then the behaviour is like the one-voice case when you play single notes in succession; but when you play a chord, each note of the chord gets its own trigger. (To count as a chord, each note needs to be within 200 msec of its predecessor.)

> **Note**: A 'trigger' is a signal that causes certain processes to execute from the beginning. These processes include the playback of audio data (such as a sample or additive/spectral data) and various modulator types, including LFO, AHDSR, MSEG, and Sequencer.
>
> In addition to the master Trigger mode described above, individual modulators have their own TRIGGER buttons. Turning a TRIGGER button off causes the individual modulator to trigger in **Legato** mode (or not at all, if it is an LFO), **regardless** of the Master Trigger mode.

Voice count field. Determines the maximum polyphony of a preset (up to 32 voices). Setting Voice = 1 makes a preset monophonic.

Portamento controls. The portamento mode can be set to **Rate** or **Time**. 'Rate' means that Alchemy glides from note to note at a fixed rate set by the **Porto** knob (so gliding a greater distance requires more time). 'Time' means that Alchemy glides from note to note during a fixed amount of time set by the **Porto** knob (so gliding a greater distance occurs at a faster rate).

Pitch-bend Up and **Down**. Determines the response to upwards and downwards pitch-bend messages. Normally, you should set a positive 'Up' value (the default is +2) and a negative 'Down' value (the default is –2).

> **Note**: that individual Sources can be set to respond to pitch-bend or ignore it via the Keytrack field on each Source sub-page. Pitch-bend is also available as a modulator (in the 'Note Property' category), giving you more individualised control over the pitch-bend response of each Source (as well as the ability to route pitch-bend messages to mod targets other than pitch).

Tuning field. Offers a vast selection of alternative tunings. Each tuning is defined by a '**.tun**' file stored in Libraries/Tuning, within the Alchemy data folder. You can create your own TUN files with the help of Manuel Op de Coul's free *Scala* software.

18.3 The PRESERVE button

You can 'lock in' various Master parameter settings with the **PRESERVE** button, so that their current values are preserved when new presets are loaded. For instance, if you are playing live with other musicians and everyone is tuned to a lower or higher reference pitch, you can adjust the fine tuning of Alchemy to match the rest of the group and then click **PRESERVE**. This will ensure that the proper fine tuning remains in effect as you load new presets.

Parameters affected by **PRESERVE** are: Pitch-bend Up/Down, Tuning, Coarse & Fine Tune.

19 Modulation

Alchemy features a unique 'modular' modulation system which combines ease of use with almost limitless flexibility. It's a bit different from an ordinary synthesiser's modulation system, but don't worry: you can learn the basics in no time.

19.1 Working with modulators

The modulation section occupies the centre portion of the display. This area is updated dynamically to display modulation sources for the active control – indicated by a **blue circle** in the centre of the target control. In the following image we can see the Master Amp **mod rack** entries, with the controls for envelope 'AHDSR 1' visible for editing.

Almost every knob on the Alchemy interface is a **mod target**, representing a parameter that can be modulated by as many as five modulators.

When a knob has one or more modulations applied to it, a **green mod arc** is illuminated alongside the knob's blue value arc. This confirms that the knob is a mod target, and also indicates the effective **range** of the modulation.

To view or edit the modulations assigned to a parameter, normally you just have to click the parameter's knob. The parameter name is shown in the **Target** field at the top left of the Mod section; directly below it, you'll find a **mod rack** with five slots. Click a different knob and the mod rack will be updated accordingly.

To **select**, **create** or **change** a modulator, click a slot in the mod rack and choose from the pop-up menu that appears.

To **undo** a modulation assignment, choose **None** from the pop-up menu. The modulator will no longer affect the currently selected target (although it will still be available for assignment to other parameters).

To **delete** a modulator entirely (so that it no longer affects any parameters), choose 'Del' from the appropriate sub-menu (e.g. to delete LFO 2, click a slot and in the pop-up menu choose 'LFO' > 'Del LFO 2').

You can also work with modulators by **right-clicking** a target knob and choosing a command from the context menu that appears.

- **Add modulation** lets you assign a new modulator by selecting it from the appropriate sub-menu. The new modulation will appear in the first empty slot of the mod rack.
- **Clear modulation** removes all modulations from the knob, leaving the mod rack empty.
- **Copy modulation** places information on the clipboard about all the currently assigned modulators.
- **Paste modulation** applies all the modulator information from the clipboard. Using the 'Copy' and 'Paste' commands, you can quickly assign the same modulations to multiple targets.

> **Note**: the most recently clicked knob is highlighted with an illuminated spot in the centre. This makes it easy to see at a glance which knob represents the current target. If you switch between sub-pages in the interface (e.g. from the **Source A** to the **Source B** sub-page), the highlight will shift to the corresponding knob on the new page, so that you can quickly set up modulations to a recurring parameter such as Fine Tune for multiple sources.

19.2 Types of modulator

The available modulator types are as follows:

- LFO
- AHDSR
- MSEG
- Sequencer
- Note Property
- Perform

Each modulation type is described on its own page.

19.3 Modulation rack controls

Each slot in the mod rack has its own **ON** button (for toggling the loaded modulator on or off) and its own **Depth** control (adjustable from -100% to 100%). Double-clicking a Depth control resets it to zero.

Each slot also has a button marked **E** (for Edit). Clicking **E** causes the selected modulator's control panel to be displayed in the right-hand half of the MOD section. Click a different modulator's Edit button and the display updates accordingly.

Between the **E** button and **Depth** control, each slot also has a pop-up menu where you can assign, create or delete a ModMap. The ModMap module is described on its own page.

Finally, a **Smooth** knob is available at the top of each mod rack. At the default value of zero, Smooth has no effect. At higher values, Smooth causes the target to respond more gradually to modulation. If you find that rapid Filter or Amp modulation causes unwanted clicks in your sound, one solution is add a small amount of smoothing via this knob. Larger values of smoothing also produce interesting effects.

19.3.1 Modulator knobs are different

Normally, two things happen as soon as you click a knob. First, the mod rack associated with the clicked knob is displayed at the left of the Mod section. And second, if a modulator has already been assigned to the first slot in the newly displayed rack, then its control panel appears in the right-hand half of the MOD section.

But in two cases, Alchemy prevents these things from happening. If the knob you click is located on a modulator control panel (for instance, if you've clicked LFO Rate or AHDSR Attack Time), Alchemy assumes you want to adjust the modulation of the current target instead of switching to a new target. Similarly, if you click on a modulation Depth knob in the mod rack, Alchemy assumes you want adjust the depth setting, so it keeps the current rack (and thus the knob you've just clicked) in view.

You can still view and edit modulation assignments in these cases. If a knob is located on a modulator control panel or in the mod rack, simply **right-click** it to access a contextual menu with all the usual commands plus a new one: **Edit Modulation**. Selecting Edit Modulation brings the knob's mod rack into view immediately. For more details, see the examples below.

Note that a few knobs **cannot** be modulated:

- The **Smooth** knob at the top of each mod rack is not a mod target.
- You can **modulate mod depth**. However, you cannot modulate the depth of modulation of a mod depth - which is just fine, because it would take more sentences like this one to explain it if you could!
- **Knobs** in the **Perform Section** are not modulation targets; they are intended as **sources**, rather than targets, of modulation.

If a parameter is set via a menu selection rather than a knob (e.g. **Loop mode**, **Filter type**), then you cannot modulate it.

Apart from these few exceptions, if it's a knob, you can modulate it in Alchemy. This includes additive/spectral/granular play position, AHDSR times and levels, and numerous Effects parameters. Try them and see!

19.3.2 Finding a modulator's targets

The mod rack shows you at a glance all the modulators that are applied to a particular target. But consider the reverse situation: how do you find all the targets of a particular modulator? This information is available via the **Target** button, which is located at the **top right** of the **Mod section**.

Clicking the **Target** button opens a pop-up menu listing all the modulators currently in use, and all the targets to which each modulator is assigned. For example, if you initialise Alchemy (by clicking the FILE button and choosing 'Clear' from the pop-up menu), then you can browse the Target menu to see quickly that:

- AHDSR 1 modulates Master Amp
- Velocity (a Note Property) modulates Master Amp
- No other modulations have been assigned

19.3.3 A very simple modulation example

In Alchemy's Title Bar click **FILE** and choose 'Clear' to initialise the preset. Play a note and you'll hear the familiar default sawtooth wave.

In the Source section, click the **Amp** knob for Source A. In the Mod section, the **Target** field will change to read 'Amp A'.

Click in the top slot of the rack and from the pop-up menu choose 'LFO' and then 'LFO 1'. Click the **LFO** button to display the LFO controls in the field to the right.

> **Note**: Source A's **Amp** knob is now surrounded by an illuminated green band. Read Working with modulators (above) to find out why!

Hold a note and you'll hear a 'tremolo' effect, as Source A's amplitude is modulated by LFO 1. Change the LFOs Rate setting to adjust the speed of the effect.

Try adding another modulator to the rack (in the slot beneath where LFO 1 is loaded) and see what effect that has.

> **Note**: Up to **five** different modulators of any type can be assigned to **each control** in a preset.

19.3.4 Modulating LFO Rate

In the preceding example, we created a 'tremolo' effect by modulating Source A's **Amp** knob with LFO 1. Next, we'll modulate the **Rate** of this LFO with **KeyFollow**, so that the tremolo effect is faster for higher notes.

> **Note**: you can adapt this example to work with other modulator parameters, such AHDSR Attack time.

Right-click LFO 1's Rate knob and choose Add Modulation > Note Property > KeyFollow.

'LfoRate' now appears in the Target field, and KeyFollow is listed in the first slot of the mod rack.

Play a variety of higher and lower notes to hear the effect of KeyFollow on the LFO Rate. If this effect is too extreme, you can reduce the Depth setting in the first slot of the mod rack.

You can read more about the KeyFollow modulator on the Note Property page.

Modulating modulation depth

Each modulator has its own **Depth** control, and each **Depth** control can itself be assigned a modulator.

> Note: a modulator can even be set to modulate its own **Depth**!

Right-click a **Depth** control in the modulation rack.

From the pop-up menu that appears choose 'Edit Modulation'.

The **Depth** control's name will appear in the **Target** field, e.g. 'Mr1Depth' (indicating that the target is Mod rack, slot 1 Depth).

Add one or more modulators to the rack in the usual way.

> **Note**: that the Modwheel is not available directly as a modulator. Instead, the Modwheel is always linked with one of the Perform Controls. If you are starting from an initialised preset, you can assign the Modwheel to control a mod Depth by choosing Perform > Control7 as a modulator.

19.4 LFO

The **LFO** module provides a standard Low Frequency Oscillator, such as you might find in any conventional synthesiser or sampler. There are one or two unusual features, however.

TRIGGER. When the **TRIGGER** button is activated, the LFO re-triggers (starts again from zero) with each new note. With **TRIGGER** deactivated, the LFO is free-running.

FILE. Clicking the **FILE** button opens a pop-up menu from which LFO presets can be loaded or saved to files (*.lfo). A collection of useful presets is included (in Alchemy/Libraries/Lfo). You can also 'Copy' and 'Paste' settings between LFOs, or choose 'Clear' to initialise the module.

BIPOLAR. When the **BIPOLAR** button is activated the LFO outputs both negative and positive values in each cycle (from -50% to 50%). With **BIPOLAR** deactivated, only positive values are output (from 0% to 100%).

Current LFO field. Alchemy provides up to 16 LFOs (one by default, more if you create them when assigning modulators — see the Modulation page of this Manual for details). You can access each LFO's control panel by selecting its number in the Current LFO field.

Shape. Click the **Shape** field to choose an oscillator waveform for the LFO. There are several categories:

- 'Basic' contains the most familiar choices, such as 'RampUp', 'RampDown', 'Sine', 'Square', and 'Triangle'.
- 'Basic' also contains two randomised choices. Use 'RandGlide' when you need a ***constantly fluctuating random modulator***; it reaches new random values at a speed set by the Rate control, and it glides smoothly from one value to the next. Use 'RandHold' when you need a ***stepped random modulator*** (e.g. for ***sample and hold*** effects); it jumps to new random values at a speed set by the Rate control, holding each value until the next jump occurs.

The remaining categories offer more complex shapes; it takes a bit of trial-and-error to learn which shapes are suitable for a particular purpose. The 'Serial-Angular' category offers a variety of 'stepped' shapes, while the 'Serial-Smooth' category provides various complex rising-and-falling patterns. Finally, the 'UHF' (ultra-high frequency) category offers shapes containing multiple copies of some pattern, such that the effective modulation speed is a multiple of the speed set by the Rate control.

> **Note**: See the LFO Shapes section later in the manual for a graphic representation of the LFOs installed with Alchemy.

Rate. Controls the LFO rate or frequency. When the **Sync** button is deactivated, the rate is adjustable over a range of 0 Hz to 220 Hz. With **Sync** activated, tempo information from the host application is used to calibrate the **Rate** control in rhythmically meaningful units (i.e. beats and fractions of a beat).

Sync. Determines whether the Rate control (see above) is set in Hz (Sync off) or in Beats (Sync on).

Delay. When **TRIGGER** (see above) is activated, **Delay** introduces a delay between the note-on message and the first cycle of the LFO. The delay is adjustable over a range of 0.00 seconds to 20.00 seconds. When **TRIGGER** is deactivated, **Delay** has no effect.

Attack. When **TRIGGER** (see above) is activated, **Attack** applies an envelope to the LFO's output, effectively 'fading it in', so that the modulation depth steadily increases the longer a note is held. The attack time is adjustable over a range of 0.00 seconds to 20.00 seconds. When **TRIGGER** is deactivated, **Attack** has no effect.

Phase. When **TRIGGER** (see above) is activated, **Phase** allows the starting point of the LFO to be adjusted from 'zero' to later in the cycle. The available range is 0.00% to 100.00%.

> **Note**: the LFO module has no depth control. The modulation depth is adjusted using the **Depth** knob alongside the module's slot in the modulation rack.

19.5 AHDSR envelopes

The **AHDSR** module provides an envelope generator with Attack, Hold, Decay, Sustain, and Release stages. As pictured above, the A, H, D, and R stages have settable times, while the S stage has a settable level (which is maintained until a Note Off occurs). This is a fairly standard type of modulator, although the implementation in Alchemy is more versatile than most.

Current AHDSR field. Alchemy provides up to 16 AHDSRs (one by default, more if you create them when assigning modulators — see the Modulation page of this Manual for details). You can access each AHDSR's control panel by selecting its number in the Current AHDSR field.

FILE. Clicking the **FILE** button opens a pop-up menu from which envelope presets can be loaded or saved to files (*.ahd). A collection of useful presets is included (in the 'Libraries' folder within Alchemy's data directory). You can also 'Copy' and 'Paste' settings between AHDSR modules, or choose 'Clear' to initialise the module.

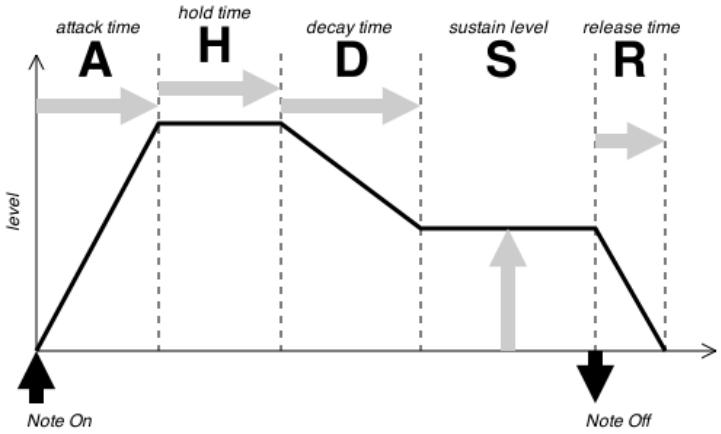
TRIGGER. When the **TRIGGER** button is activated, the envelope re-triggers (starts again from the beginning) with each new note. With **TRIGGER** deactivated, the envelope is effectively monophonic (i.e. it triggers on the first note and persists for subsequent notes in the same legato phrase).

Attack. Controls the envelope's 'attack' time, i.e. the time taken to reach peak amplitude after a note is played. From 0.00 seconds to 20.00 seconds.

Hold. Controls how long peak amplitude is held before the 'decay' stage of the envelope begins. From 0.00 seconds to 20.00 seconds.

Decay. Controls the envelope's 'decay' time, i.e. the time taken for the amplitude to ramp down to the 'sustain' level. From 0.00 seconds to 20.00 seconds.

Sustain. Sets the envelope's sustain level, as a percentage of peak amplitude (adjustable from 0% to 100%).

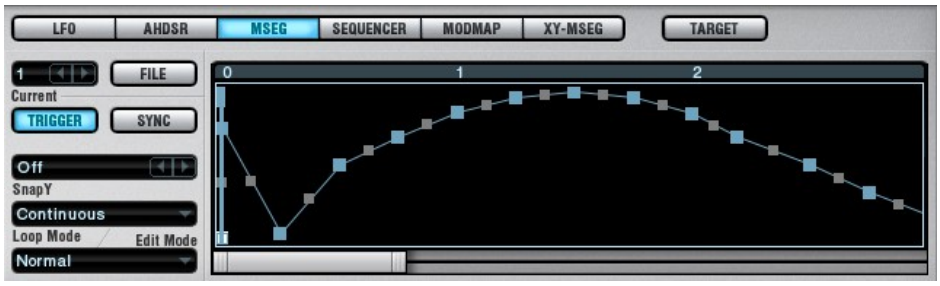


Release. Controls the envelope's 'release' time, i.e. the time taken for signal amplitude to ramp back down to zero. From 0.00 seconds to 20.00 seconds.

19.5.1 Envelope shaping

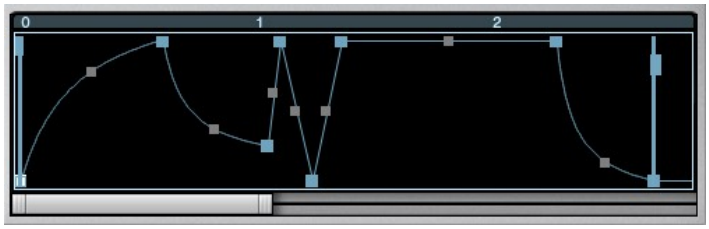
The **Attack**, **Decay** and **Release** controls each have an associated set of **switches**, giving you a choice of **shapes** for each of these stages. There is a linear shape; two **convex** shapes, with different degrees of steepness; and two **concave** shapes, again with different degrees of steepness. The different shapes produce characteristically different effects.

19.6 MSEG



The MSEG module provides a sophisticated **M**ultiple **S**egment **E**nvelope **G**enerator that allows complex modulation envelopes to be created and edited.

Any number of **breakpoints** can be added to an envelope. The envelope segments linking these breakpoints can be linear or curved. A **Sync** function allows envelopes to be anchored to a grid derived from the host application's tempo to create elaborate rhythmic patterns.



The MSEG display shows a graph of the envelope generator's output. The ruler along the top shows the time, calibrated in seconds (or in beats when Sync is activated - see below). The envelope appears as series

of small squares (breakpoints) joined by lines/curves representing the different envelope segments. A **Play Cursor** scrolls across the display, tracking the progress of the envelope relative to the most recently played note.

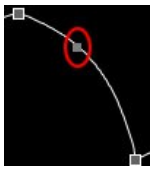
19.6.1 Editing MSEG envelopes

Envelopes are created and edited in two basic ways: by adding, moving or removing breakpoints, and by adjusting the curve of the envelope segments between breakpoints.

To **move** a breakpoint, simply click and drag it.

To move a several adjacent breakpoints together **as a group**, first drag around the breakpoints to select them all. Then click and drag any breakpoint in the group.

To **add** a breakpoint, right-click (command-click) at the place where you want the new breakpoint to appear.



To **remove** a breakpoint, right-click (command-click) it.

To adjust the **curve** of a segment, click and drag the segment handle

To **reset** the curve of a segment, double-click the envelope segment

A breakpoint's ***x-axis*** value (time position) and *y-axis* value (level) are reported together in the parameter value display when you click the breakpoint, and the displayed values update as you drag the breakpoint.

Each envelope segment has a 'handle' (a square box, slightly smaller than a breakpoint) halfway along its length, which is used to adjust the curvature of the segment. Double-clicking the handle will reset the segment to a linear slope.

Dragging the handle upwards causes the segment to become progressively more convex; dragging it downwards causes the segment to become progressively more concave. Convex, 'flat' and concave envelope segments produce characteristically different effects.

19.6.2 Edit modes

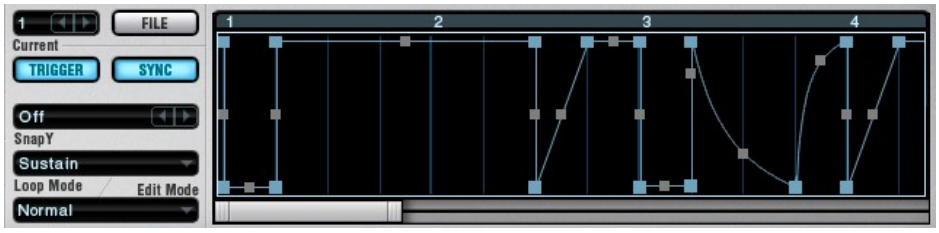
The **Mode** pop-up menu allows you to choose from one of three editing modes that determine how the envelope will react to being edited.

Normal. One breakpoint at a time can be moved, by clicking and dragging. The surrounding points remain stationary.

Slide. Dragging a breakpoint also moves all subsequent points in the envelope, so that the relative distance between these points is preserved.

Stretch. Dragging a breakpoint to the left compresses earlier points and stretches later points, while dragging it to the right stretches earlier points and compresses later points. In either case the total length of the envelope is preserved.

19.6.3 Sync Mode



When the **Sync** button is activated, the time ruler along the top of the MSEG display is recalibrated in beats (calculated from the host application's current tempo) and a grid of vertical lines (spaced at quarter-beat intervals, or less frequently when zoomed out) is superimposed. Breakpoints **snap** to these grid lines when dragged or created, making it easy to create precisely-aligned rhythmic envelopes.

19.6.4 Loop markers

Two pale blue vertical lines also appear in the MSEG display, each with a small rectangular 'handle'. These are the envelope loop markers.

> **Note**: if 'None' is selected in the **Loop** pop-up menu (see below) the envelope has no loop, and although the loop markers will remain visible they have no effect.

The marker with the higher, leftward handle controls the loop **start** point. The marker with the lower, rightward handle controls the loop **end** point.

The loop markers are moved by simply dragging them in the desired direction. For obvious reasons, the loop start marker cannot be moved to the left of the loop end marker, and the loop end marker cannot be moved to the right of the loop start marker. Loop markers always snap to the nearest breakpoint.

The loop modes, accessible from the **Loop** pop-up menu, are as follows:

None. Looping is disabled.

Continuous. The looped section plays continuously in a forward direction while a note is held, and goes on looping after the note is released.

Sustain. The loop section is played while a note is held. When the note is released the remainder (or release section) of the envelope plays.

Forward/Back. Like 'Continuous' except that the looped section is played alternately forwards and backwards.

19.6.5 Other controls and parameters

SnapY quantises the breakpoint levels (or *y* values), limiting them to exact fractions of the available range. For instance, a SnapY setting of '1/3' means that breakpoint levels will snap to the values 0, 1/3, 2/3, and 1 when dragged. A SnapY setting of 'None' turns quantisation off and allows you to set breakpoint levels freely.

> **Note**: that the SnapY setting does not move existing breakpoint levels into alignment with quantised positions; it only affects how breakpoints respond when created or dragged.

Current MSEG field. Alchemy provides up to 16 MSEGs (two by default), and you can access each MSEG's control panel by selecting its number in the Current MSEG field.

FILE. Clicking the **FILE** button opens a pop-up menu from which you can **Load**/**Save** MSEG presets from/to files (*.mse). A collection of useful presets is included (in the 'Libraries' folder within Alchemy's data directory). You can also **Copy** and **Paste** settings between MSEGs, or choose **Clear** to initialise the module.

TRIGGER. When the **TRIGGER** button is activated, the envelope re-triggers (starts again from the beginning) with each new note. With **TRIGGER** deactivated, the envelope re-triggers for the first note only and continues its progress for subsequent notes: the MSEG will only re-trigger for notes received after all other notes have completed the release phase.

19.6.6 Example: Modulating pitch

First, **initialise** the preset (by clicking **FILE** in Alchemy's Title Bar, and choosing 'Clear').

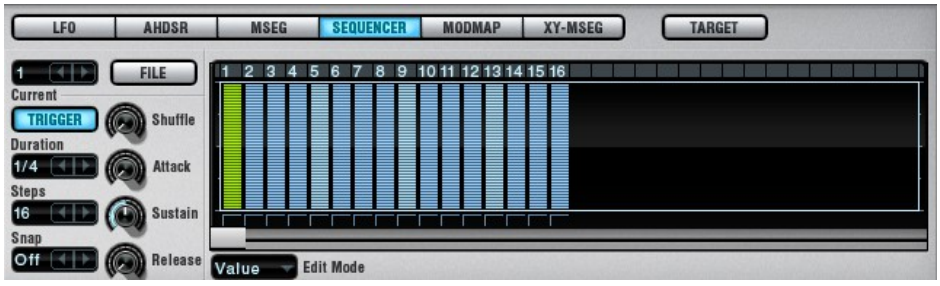
In the Master section, click the **Coarse-Tune** knob, and assign 'MSEG 1' as its modulator.

In the Mod Rack, set MSEG 1's **Depth** control to 24 semis.

If the MSEG module's control panel is not already in view, click the **MSEG** button at the top of the Modulation section to access it. From the SnapY pop-up menu, choose '1/24'.

Now an MSEG envelope can be used to control Alchemy's pitch in semitone increments, over a two-octave (i.e. 24-semitone) range.

19.7 Sequencer



The Sequencer module provides a powerful step-based modulator that synchronises with the host tempo and can be programmed with patterns of up to 128 steps. In addition to the level of each step, its groove/shuffle and its envelope can be controlled, both globally for the whole pattern and locally per step.

The step editor at the right-hand side of the Sequencer module is fairly self-explanatory. Steps are numbered from left to right across the top of the display. In the usual **Values mode**, the value of each step is represented by a vertical bar (or by a totally empty column when the value is zero).

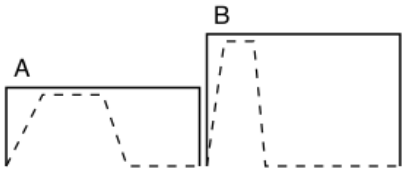
Drag a bar up or down to adjust its value, or click directly at the desired height.

Set the values of multiple bars by simply **dragging left** or **right** across them.

Right-click (or control-click) a bar to set it to **zero**.

Tie one step to the next by shift-clicking below it. A small 'chain-link' symbol appears below each tied step. When two or more steps are tied together, they behave like one longer step.

The step editor can be switched to two additional modes besides the usual Values mode. In **Length mode**, you can edit the step lengths to create a pattern of longer and shorter envelope shapes (so that the attack, sustain, and release stages fill a larger or smaller portion of the fixed step duration). These lengths are combined with the Sequencer's overall **Attack**/**Sustain**/**Release** settings to determine the envelope shape of each step. In the following illustration, Step B has a greater *value* than Step A, but Step A has a greater *length* than Step B.



When the step editor is in **Swing mode**, finally, you can create variations in the timing of the steps. Each swing value ranges from 0 to 2; the middle value of 1 represents normal timing, while smaller values play earlier and larger values play later than normal. These swing values are combined with the overall timing pattern determined by the Sequencer **Shuffle** control.

The remaining Sequencer controls are as follows:

FILE button — click to access a pop-up menu with the following commands:

- **Save** stores the current Sequencer configuration (step values, lengths, and swing settings; plus the settings of the remaining Sequencer controls) in a new disk file (*.seq), while **Load** applies the configuration stored in an existing disk file.
- **Copy** stores the current Sequencer configuration on the clipboard, and **Paste** applies the configuration currently found on the clipboard.
- **Clear** resets the Sequencer controls and pattern to their default settings.
- **Randomise** applies random offsets to the Shuffle, Attack, Sustain, and Release settings (described below).
- The **Import** functions are described in a separate section below.

Current Sequencer field — Alchemy provides up to 16 Sequencers, and if you are working with a preset that uses more than one, you can access each Sequencer's control panel by selecting its number in the Current Sequencer field.

TRIGGER button — When TRIGGER is activated (lit) the sequencer pattern plays from the beginning with each MIDI note-on. With TRIGGER deactivated (unlit), the Sequencer runs continuously.

> **Note**: the Sequencer module always play in sync with the host tempo, regardless of the **TRIGGER** setting.

Duration determines the duration of each and every step in the pattern, expressed as a fraction of a beat. (Assuming a quarter-note beat, '1/2' produces eighth note steps and '1/4' produces sixteenth note steps.)

Steps determines the length of the Sequencer pattern (number of steps, from 1 to 128)

Snap quantises the step values, limiting them to exact fractions of the available range. For instance, a Snap setting of '1/2' means that bars will snap to the values 0%, 50%, and 100% when dragged. A Snap setting of 'None' turns quantisation off and allows you to set step values freely.

> **Note**: that the **Snap** setting does not move existing step values into alignment with quantised positions; it only affects how step values respond when you move them by clicking and dragging in the step editor.

Shuffle lets you create various 'swing' effects. Setting a Shuffle value greater than 0% increases the duration of the odd-numbered steps (1, 3, 5, …) and decreases the length of the even-numbered steps correspondingly.

Attack determines the amount of time each step takes to reach its peak level.

Sustain determines the amount of time each step is held at its peak level.

Release determines the amount of time each step takes to fall from its peak level.

19.7.1 Importing from a MIDI file

Alchemy is capable of extracting information from a short MIDI file and applying it to patterns in the step editor. Specifically:

It can extract *velocity* data and set the step values to match. If the MIDI file consists of notes of equal duration (e.g. a succession of eighth notes, or a succession of sixteenth notes), then every step in the resulting pattern will have an associated non-zero value. If the MIDI file consists mainly of notes of equal duration with

occasional gaps (e.g. a succession of eighth notes with occasional eighth rests), then the gaps will be represented by step values of zero. If the MIDI file has more irregular timing, or if it consists of chords rather than single notes, then the results of this process will be less predictable and usually less useful.

It can extract *groove data* (timing inflections) and set the step swing values to match. If the MIDI file consists of nearly equal durations (e.g. eighth notes or sixteenth notes with timing inflections), this process yields meaningful results.

It can extract *note data* (pitches) and set the step values to match. The MIDI file should consist of equal durations with no gaps, and it should be limited to single pitches between a low C and a C two octaves higher (e.g. C1 through C3); the low C corresponds to a step value of zero. (Because the pitch range is always two octaves, which is equivalent to 24 semitones, you should use the sequencer to modulate pitch with a depth of 24 semitones in order to reproduce the pattern of notes in the original MIDI file.)

Three import commands in the Sequencer **FILE** pop-up menu make use of these capabilities.

Import Velocity — sets step values based on extracted velocity data and swing values based on extracted groove data.

Import Note — sets step values based on extracted note data and swing values based on extracted groove data.

Import Groove — sets swing values based on extracted groove data.

19.8 Note Property

Several properties of incoming MIDI note data, as well as values generated per-note by Alchemy, are available as modulation sources.

> **Note**: unlike the other modulators, Note Property has no control panel to display.

Click a slot in the modulation rack and choose 'Note Property' from the pop-up menu. A sub-menu appears, from which the following options are available:

Velocity. Modulation based on the velocity values of incoming MIDI note data.

KeyFollow. Modulation based on incoming MIDI note numbers (i.e. the modulation value increases as you play higher pitches on your MIDI keyboard controller). This is a bipolar source, with C3 corresponding to zero.

ChanAftertouch. Unipolar modulation based on channel aftertouch data.

PolyAftertouch. Unipolar modulation based on poly after touch data.

> **Note**: Channel aftertouch is transmitted by many MIDI controllers; it consists of one variable stream of values per MIDI channel. Poly aftertouch, which consists of a variable stream of values per individual note, is a much less common feature. If your controller doesn't produce poly aftertouch (or channel aftertouch, for that matter), you can probably still create data of this type directly in your sequencer software.

Speed. Modulation based on the elapsed time between notes (e.g. a progressively slower sequence of notes results in progressively greater modulation values).

Held. A modulation signal that rises to full-scale immediately at note-on and falls to zero immediately at note-off.

FlipFlop. A modulation signal that is alternately full-scale and zero on successive notes.

FlipFlop2. LikeFlipFlop, but the value reverses every two notes: zero, zero, full, full, repeat.

> **Note**: The FlipFlop modulators can be used together to create a 'round-robin' involving all four of Alchemy's Source modules. With the Morph mode set to 'Morph XY' or 'XFade XY', set the Morph X and Y knobs both to 0%; then modulate X with FlipFlop and Y with FlipFlop2 (or vice versa). (See the Morph page for information on morphing and cross-fading.)

Random1- **Random4**. Modulation based on a fixed random value per note. This is a unipolar source with values ranging from zero through full-scale. The four sources of this type are randomised independently of one another.

PitchBend. Modulation based on MIDI pitch-bend messages. This is a bipolar source with values ranging from negative full-scale through positive full-scale.

Max. Modulation based on a constant full-scale value.

ChanAftertouchBipolar. Bipolar modulation based on channel aftertouch data.

PolyAftertouchBipolar. Bipolar modulation based on poly aftertouch data.

> **Note**: the bipolar Chan- and Poly- aftertouch modulation sources start with half of the full modulation amount applied to the target parameter. Incoming MIDI aftertouch data modifies the initial amount by +/- 50%. This contrasts with the unipolar Chan- and Poly- aftertouch modulation sources which have an initial value of 0, increasing towards 100% as aftertouch is applied.

19.9 Perform

When you click a slot in the modulation rack and choose the 'Perform' sub-menu, you can select from a list of sixteen different modulation sources.

These correspond to the sixteen different controls (eight assignable knobs, two X/Y control squares, four envelope knobs) available in Alchemy's Perform section. For more information on working with these controls, see the Performance Controls page.

The available modulations sources are:

Control1 – **Control 8**. Assignable knobs, numbered 1–8.

XYPad1X. The X (horizontal) axis of the left-hand XY control square.

XYPad1Y. The Y (vertical) axis of the left-hand XY control square.

XYPad2X. The X (horizontal) axis of the right-hand XY control square.

XYPad2Y. The Y (vertical) axis of the right-hand XY control square.

MAttack, **MDecay**, **MSustain** and **MRelease**. The **Attack**, **Decay**, **Sustain** and **Release** knobs.

19.10 ModMap

A **ModMap** is not a modulator. Instead, its purpose is to **process the output of a modulator**, mapping the original values to new ones before they are applied to a particular modulation target. With ModMaps, you can create 'curved' velocity responses, scale the volume of each Source across the keyboard, 'quantise' the pitch response to random-LFO modulation so it adheres to the steps of a scale, and much more.

> **Note**: ModMaps are also used to define the response curves of Additive and Spectral Formant Filters. See the description of the Additive Formant Filter for more details.

This page is organised in three parts. First, it explains how to assign a ModMap to a modulation in one of Alchemy's mod racks. (Please see the Modulation page for a broader discussion of the mod rack system.) Second, it provides a number of 'recipes' for ModMaps that perform common functions; you can use these as-is or adapt them for your needs. And third, it offers a more complete explanation of how ModMaps work — the details you'll want to know when you design ModMaps of your own.

19.10.1 Assigning a ModMap

The **modmap field** at the far right of each mod-rack slot displays the number of the ModMap, if any, that currently applies to the modulation set up in that slot. A dash '–' indicates that no ModMap has been assigned.

To **select**, **create**, or **change** a ModMap, click in the modmap field and choose from the pop-up menu that appears.

To **undo** a ModMap assignment, chose the dash '–' from the pop-up menu. The ModMap will no longer affect the modulation set up in this mod-rack slot (although it will still be available for assignment to other modulations).

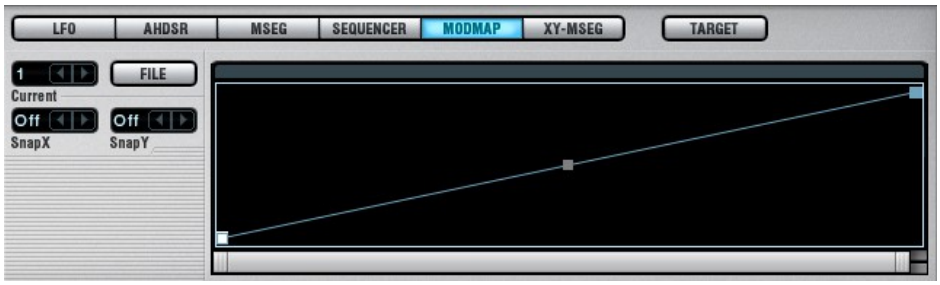
To **delete** a ModMap entirely (so that it no longer applies to any modulations), choose 'Del ModMap' from the pop-up menu.

19.10.2 ModMap recipes

These examples illustrate typical uses of the ModMap feature. For an explanation of why each ModMap works the way it does, read *the details* at the bottom of this page.

The default ModMap

Newly initialised presets in Alchemy have a single ModMap shaped as follows.



Each additional ModMap that you create begins with this same default shape — a straight line increasing from 0.00 at the far left to 1.00 at the far right. This **default ModMap does nothing** — it 'maps' modulations values by passing them through unchanged.

Although it doesn't do anything useful on its own, the default ModMap is a starting point for all the recipes that follow.

Velocity curve

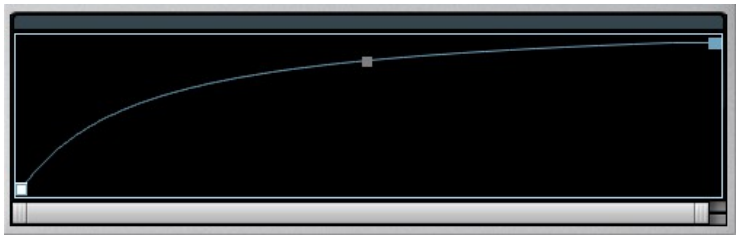
Initialise Alchemy by choosing the **Clear** command in the Title Bar **FILE** menu. By default, an initialised preset has Velocity assigned as a modulator to Master Amp, so when you strike keys harder or softer (on a velocity-sensitive MIDI keyboard), notes play louder or softer.

Click on the **Master Amp** knob to bring its mod rack into view in the MOD section. You'll see two modulators loaded in the modulation rack: **AHDSR 1** in the first slot, **Velocity** in the second.

Click the dash '–' in the **modmap field** at the far right of the 'Velocity' slot, and choose '1' from the pop-up menu. This applies the default ModMap to Velocity modulation of Master Amp, and it brings the modmap controls and editor into view in the right-hand half of the Mod section. Play a few notes on your MIDI keyboard to confirm that the velocity response is unchanged by the default ModMap.

The **default ModMap** (pictured above) consists of a single segment with a breakpoint at each end and a handle in the middle.

Click on this **handle** and **drag upwards**. The segment will curve and become convex, as in the image on the right.



Play a few more notes on your MIDI keyboard and notice the effect of the convex velocity curve: notes you strike with medium force play louder than they did with the default curve.

Now **click** on the same mid-segment handle and **drag downwards** until the segment curves in the opposite direction, becoming concave. Play a few more notes on your MIDI keyboard and notice the effect of the concave velocity curve: notes you strike with medium force play softer than they did with the default curve.

Scaling volume across the keyboard

Initialise Alchemy by choosing the **Clear** command in the Title Bar **FILE** menu. From the Source All sub-page, Load one sample (or multi-sample) into Source A, and load a contrasting sample (or multi-sample) into Source B. By default, there is a 50% crossfade between these sources, so you should hear a balanced mix of the two samples across the entire keyboard. (For details about loading samples, see the Source page.)

> **Note**: that Alchemy plays loaded samples in Granular mode by default. For ordinary sample playback, you may prefer to switch Sources A and B from Granular to Sampler mode. See the Granular page for details.

Suppose that we want both Sources to play at full strength from the bottom of the keyboard up through approximately C4, and that above this point we want Source A to remain at full strength while Source B gets softer as we go further up the keyboard. Since we want volume scaling on Source B only, we will operate on the Source B Amp knob rather than the Master Amp knob. (The Source B Amp knob is accessible from either the Source All sub-page or the Source B sub-page.)

Click the Source B Amp knob to bring its mod rack into view in the Mod section. In the first slot of the mod rack, choose Note Property > KeyFollow. Now the amplitude of Source B is modulated according MIDI note number, but this modulation does not yet have the shape we want across the keyboard. Currently, the lowest notes are the softest, notes in the middle of the keyboard are medium-loud, and notes at the top are the loudest. Now we will use a ModMap to reshape the response of Source B Amp to modulation by KeyFollow.

At the far right of the first slot of the Source B Amp mod rack, click the dash '–' in the modmap field and choose '1' from the pop-up menu. This applies the default ModMap to KeyFollow modulation of Source B Amp, and it brings the modmap controls and editor into view in the right-hand half of the Mod section. Play a few notes on your MIDI keyboard to confirm that the KeyFollow response is unchanged by the default ModMap.

In order to produce the desired KeyFollow response, edit the ModMap shape. The required shape is pictured below. Raise the leftmost breakpoint up to a level of 1.00. Create a new breakpoint by right-clicking (control-clicking) somewhat more than halfway from left to right; leave the level of the new breakpoint at 1.00. Finally, drag the rightmost breakpoint down to a level of 0.00. Now Source B Amp should respond appropriately as you play across the keyboard.

Quantising pitch modulation to a scale

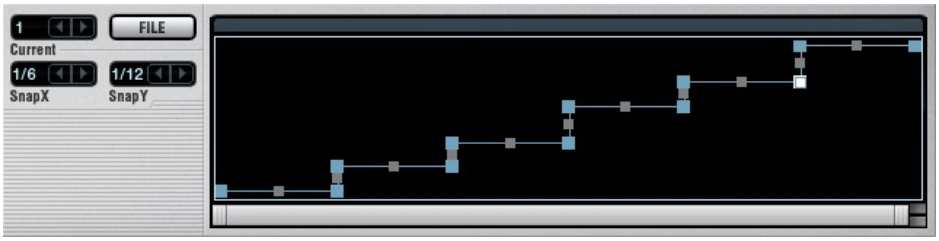
Initialise Alchemy by choosing the **Clear** command in the Title Bar **FILE** menu.

Click the **Master Coarse Tune** knob to bring its mod rack into view in the Mod section.

In the **first slot** of the mod rack, chose LFO > LFO 1, and reduce the modulation Depth to '12.0 semis'. Then adjust the LFO 1 settings as follows: Shape = 'RandHold', Rate = '1/2 beats', BIPOLAR off.

Play and hold a note to confirm that the pitch changes twice per beat.

At this point, the pitch values range freely within the specified 12.0 semitone range. The ModMap pictured below will map the LFO values to steps of a major pentatonic scale.



Use the ModMap editor's SnapX and SnapY functions to create the desired ModMap shape more easily. Set SnapX = 1/6 and SnapY = 1/12. In between the original first and final breakpoints, add ten new ones at the following **X, Y** positions:

1/6, 0/12

1/6, 2/12

2/6, 2/12

2/6, 4/12

3/6, 4/12

3/6, 7/12

4/6, 7/12

4/6, 9/12

5/6, 9/12

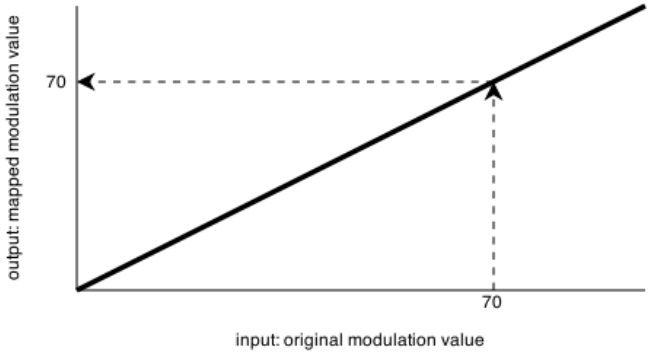
5/6, 12/12

Note: that the Y values here reflect the interval pattern of the major pentatonic scale.

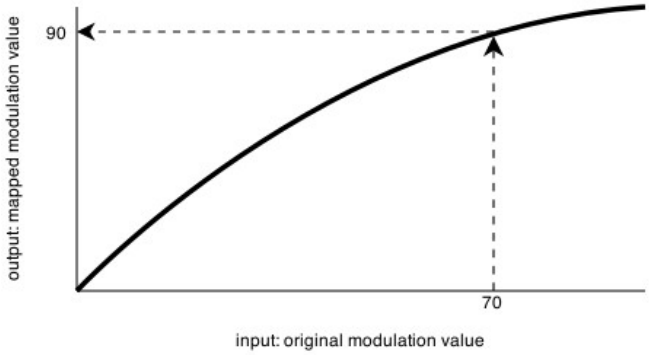
19.10.3 The details: how ModMaps work

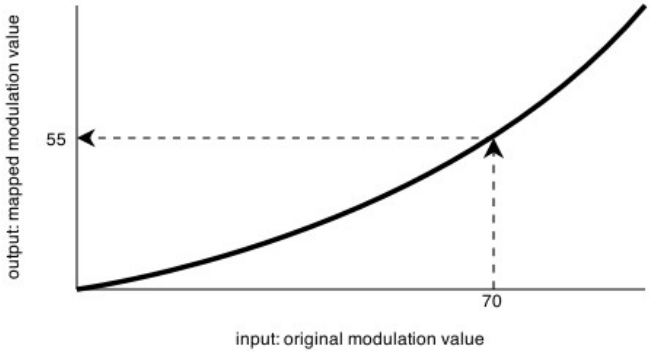
A ModMap processes the output of a modulator, mapping the original values to new ones before they are applied to a particular modulation target. The mapping is defined by the ModMap's graphical shape, which represents a **transfer function**. The x (horizontal) axis represents the range of original modulation values, from 0.00 to 1.00, while the y (vertical) axis represents the range of mapped modulation values, also from 0.00 to 1.00. To see how a modulation value will be affected by the ModMap, look up the original value along the x axis; the corresponding y value determines the output of the mapping.

The default ModMap does nothing, because the output is identical to the input.

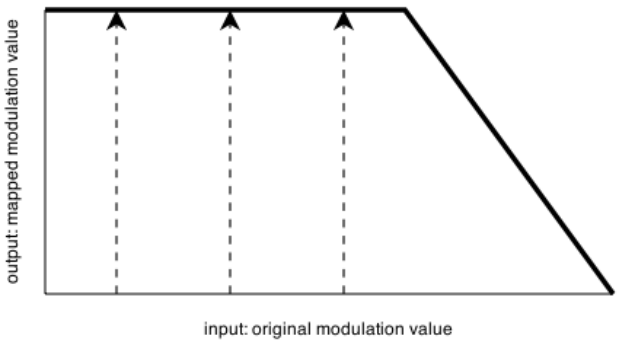


A ***convex ModMap*** maps the middle range of inputs to values that are higher than the default output, while a ***concave ModMap*** maps the same range to values that are lower than the default output.

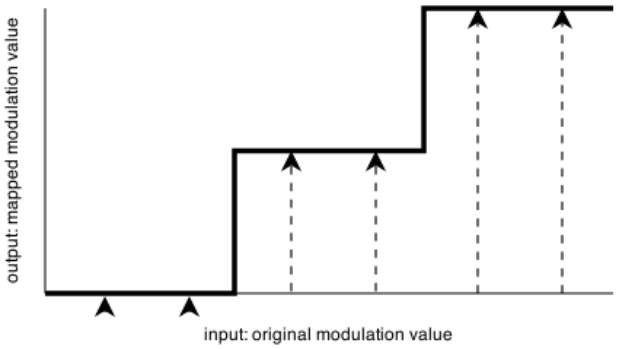




A ***ModMap with a flat (horizontal) region*** maps a range of inputs to a single output.



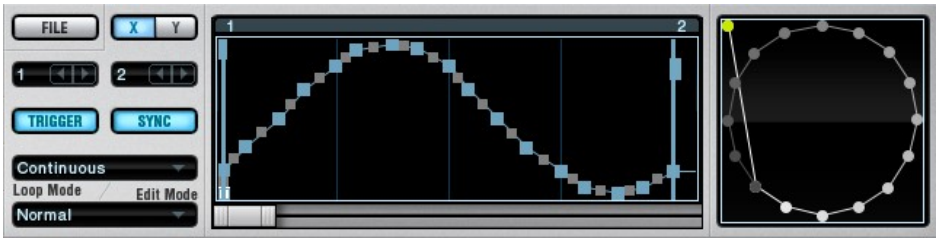
A ***stepped ModMap*** 'quantises' the input, mapping each input value to an output value defined by one of the steps.



19.11 XY-MSEG

An **XY MSEG** is not a separate type of modulator alongside Alchemy's AHDSRs, LFOs, and so on. Instead, it provides a special interface for manipulating Alchemy's regular MSEGs two at a time.

Specifically, an XY MSEG associates one MSEG with the X axis and another with the Y axis of a square, and it allows you to manipulate the levels of both MSEGs by dragging in the square. The MSEG XY editor also gives you a normal MSEG view, switchable between the X-axis MSEG and the Y-axis MSEG. In this view, you can change both the levels and the times of either MSEG. Beginning in version 1.12, a **Play Cursor** scrolls across the MSEG display, tracking the progress of the envelope relative to the most recently played note, and an associated coloured ball travels across the XY square to indicate the current XY position.



When you change either times or levels in the XY MSEG editor, you're changing the associated (regular) MSEGs. By default, MSEG 1 is selected for the X-axis and MSEG 2 is selected for the Y-axis.

To modulate a pair of controls (such as Morph X and Morph Y) with an XY MSEG, **modulate one control with the X-axis MSEG** (by default, this is MSEG 1), and **modulate the other control with the Y-axis MSEG** (by default, this is MSEG 2). See the example 'MSEG-driven morphs' at the bottom of the Morph page for step-by-step instructions.

19.11.1 Using the XY MSEG controls

The XY MSEG **FILE** button opens a pop-up menu with the following commands:

Load —Opens a dialog in which you can select a MSEG preset (*.mse) to load into the XY MSEG editor. If two presets in the same folder have identical names, but one name ends with 'X' and the other ends with 'Y' — for instance, 'Circle 1X.mse' and 'Circle 1Y.mse' — then selecting *either one* of these names loads the 'X' preset into the X-axis MSEG *and* the 'Y' preset into the Y-axis MSEG. If you select a name that is not part of an XY pair, then it will load just the selected file into the X- or Y-axis MSEG, depending on which axis button (see below) is activated.

Save — Depending on which axis button (see below) is activated, saves the X- or Y-axis MSEG (but not both) to a preset file with a name of your choice.

Copy and **Paste** — Copy places the X- or Y-axis MSEG information onto the clipboard; Paste retrieves the clipboard data and applies it to the X- or Y-axis MSEG.

Clear —Sets the X- and Y-axis MSEGs to their default configurations.

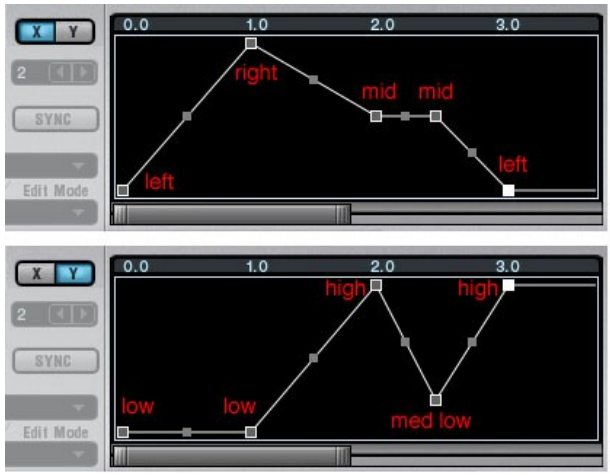
The **X** and **Y axis buttons** determine whether the X- or Y-axis MSEG is displayed and made available for editing in the breakpoint editor.

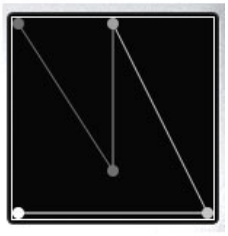
The **MSEG selection fields** determine which of Alchemy's MSEGs are associated with the X and Y axes of the XY MSEG square. The field on the left sets the X-axis MSEG, while the field on the right sets the Y-axis MSEG.

> **Note**: In order for the XY MSEG square to work properly, you should always assign two distinct MSEGs to the X and Y axes.

The **TRIGGER**, **SYNC**, **Loop Mode** and **Edit Mode** controls work just like the controls of the same names on the normal MSEG control panel (see the MSEG page for details). The one difference in the XY MSEG editor is that each of these controls applies to both the X- and the Y-axis MSEGs.

Using the XY MSEG breakpoint editor and square





Working with the XY MSEG **breakpoint editor** is just like working with the breakpoint editor on the normal MSEG control panel (see the MSEG page for details) — except for one important difference: **the number and timings of breakpoints are kept synchronised between the X- and Y-axis MSEGs**. When you **create or delete points** — by right-clicking (control clicking) — you are increasing or decreasing the number of points in both MSEGs simultaneously. Similarly, when you **drag a point to a new time position**, you are changing the timing of a point in the X-axis MSEG and in the Y-axis MSEG. (On the other hand, the *levels* of the X- and Y-axis MSEG breakpoints are independent, so changing a level in one MSEG does not affect levels in the other MSEG.)

A quick illustration

Start with an initialised preset and open the XY MSEG editor.

MSEG 1 is assigned to the X axis, and MSEG 2 is assigned the Y axis. Switch between the X and Y axis buttons to confirm that both of these MSEGs have a default shape consisting of three breakpoints.

Click the X axis button and right-click (control click) in the breakpoint editor to add a fourth breakpoint to MSEG 1 at a time position of approximately 0.50 sec.

Now switch to the Y axis view again and note that a fourth breakpoint has been added at the same time position in MSEG 2 as well.

If you have already edited one or both MSEGs using the MSEG control panel, then when you switch to the XY MSEG editor, the incoming MSEGs, which may arrive unsynchronised with one another, will immediately be synchronised — Alchemy performs this operation in such a way that the shape of both MSEGs is preserved, although the loop positions, loop mode, and trigger status of either MSEG may change.

While the XY MSEG editor's breakpoint envelope shows you the levels of either the X-axis MSEG or the Y-axis MSEG, the **XY square** shows you both at once. In the square, breakpoints are positioned such that their horizontal positions depict X values, while their vertical positions depict Y values.

A quick illustration

Start with an initialised preset and open the XY MSEG editor.

Ensure that the X axis button is active, and then click the first (leftmost) breakpoint and drag it up and down. Note that the brightest breakpoint in the square moves to the left when you drag the X point down and to the right when you drag the Y point up. *The X value controls the left-to-right position in the square.*

Now click the Y axis button to switch the view in the breakpoint envelope display, and then click the first (leftmost) breakpoint and drag it up and down again. This time, note that the brightest breakpoint in the square most to the bottom when you drag the Y point down and to the top when you drag the Y point up. *The Y value controls the bottom-to-top position in the square.*

While the XY square shows you both X and Y values, it doesn't show you the timing of each breakpoint; for that information, you'll need to refer to the breakpoint envelope display. However, the square does represent the 'chronological order' of the breakpoints by rendering them with different degrees of brightness, **from bright white for the first breakpoint through dark grey for the last**.

The XY square is not just a display, it's an editor. You can click a breakpoint in the square and drag it left/right to change its X value and up/down to change its Y value. To change the timing of a breakpoint, or to create or delete breakpoints, you should work in the breakpoint envelope display.

20 SFZ Files

The SFZ file format is used to define instruments in a way that is understood by software samplers from different developers. The SFZ standard was first developed by Cakewalk. A complete specification of the SFZ file format and the opcodes used to define instruments can be found on the Cakewalk DevXchange.

20.1 What are SFZ files?

SFZ files are plain text files with a '.sfz' file extension. The files contain 'opcodes' that determine how samples will be mapped across a standard MIDI keyboard, how each note will respond to velocity, and other attributes such as placement in the stereo field and tuning. The minimum requirements for creating or editing SFZ files are Textedit (Mac) or Notepad (Windows).

20.2 Creating your own SFZ files

Several methods may be used to create new SFZ files:

- Create a new file using a text editor
- Modify an existing SFZ file to suit your needs
- Use SFZ editor software
- Convert an existing sample library using a special utility
- Use a scripting language to generate opcodes based on the contents of a folder

> **Note**: see the SFZ page on the Camel Audio website for more information about the tools available for this purpose. In this manual we will focus on the specifics of the SFZ format relating to Alchemy and Alchemy Player.

The simplest of these methods is to copy an existing SFZ file and modify the referenced file names while taking care to ensure the key and velocity zones are correct for the desired mapping.

Creating a new SFZ file from a blank file using a text editor is practical for smaller instruments, or when combined with copy & paste and find & replace techniques for larger instruments.

20.3 Opcodes recognised by Alchemy

The following table lists the opcodes understood by Alchemy.

> **Note**: other opcodes not listed here may be present in an SFZ file. For example, after converting a sample library from some other format to SFZ. These opcodes will be ignored in the current version of Alchemy, and will not have any effect on the playback of the instrument.

Opcode	Usage Notes
`<group>`	Used to define common values applied to all regions contained within the group. Multiple groups may be defined, each containing one or more regions.
`<region>`	Used to define keyboard zones. Multiple regions may be defined.

`//`
 	
`// this is a comment`
	Comment markers; text placed after comment markers is ignored.

`default_path`
 	
`default_path=C:\Samples`
`default_path=..\Samples`
	Set the default path used to locate samples referenced without a full path specification.

`sample`
 	
`sample=Kick.wav`
`sample=..\Samples\Kick.wav`
`sample=C:\Samples\Kick.wav`
	Load the referenced sample using settings defined in the current region and group.
	
	Full path or relative path specifications may be used to reference samples.

`volume`
 	
`volume=-6`
	Specify the volume for a group or region. Value is specified in decibels. The maximum value is 0 dB.

`tune`	Specify the pitch offset for a group or region, relative to the key

`tune=-6`	value. The value is specified in semitones.

`pan`
 	
`pan=36`
`pan=-23`
	Specify the position within the stereo field for a group or region. The range is +/- 63 with negative values panning to the left; positive values pan to the right. A value of 0 is centred.

`key`
 	
`key=64`
	Specify the note that will trigger a sample. The range is 0-127.

`pitch_keycenter`
 	
`pitch_keycenter=64`
	Specify the centre pitch for a group or region. The range is 0-127. Samples above or below the centre pitch will be retuned.

`lokey`
 	
`lokey=36`
	Specify the lowest note that will trigger a group or region. The range is 0-127. Notes between pitch_keycenter and lokey will cause the sample in a region to be pitched down.

`hikey`
 	
`hikey=64`
	Specify the highest note for a group or region. The range is 0-127. Notes between pitch_keycenter and hikey will cause the sample in a region to be pitched up.

`lovel`
 	
`lovel=30`
	Specify the lowest velocity for a group or region. The range is 0-127.

`hivel`
 	
`hivel=127`
	Specify the highest velocity for a group or region. The range is 0-127.

`loop_mode`
 	
`loop_mode=0`
	The loop mode for a group or region. Allowed values are:
	
	0 = no loop 1 = continuous
	2 = sustain 3 = forward/back 4 = all

`cutoff`
 	
`cutoff=200`
`cutoff=10000`
	Specify the filter cutoff point for a group or region. The value is specified in hertz. The last used 'cutoff' value in the file is used as the value is global to the Alchemy source to which the SFZ is being imported.

`fil_veltrack`
 	
`fil_veltrack=64`
	Specify the filter cutoff keyboard tracking for a group or region. The range is 0-127. As with cutoff, this value is global to the Alchemy source.

`seq_position`
 	
`seq_position=2`
	Used to create round-robin zones, with the value specifying the position of a sample within the round-robin zone. The default value is 1. Defining a second region in the same zone with a seq_position value of 2 will cause sample playback to alternate.
	
	A value higher than the total number of round-robin samples may be used to trigger the sample multiple times before the round-robin group is reset.

`sw_last` `sw_last=64`	Used for key switching. A region will play only if the last received MIDI note number matches this value. The last played keyswitch will remain active until a new keyswitch is received.
`trigger` `trigger=attack`	Used to determine whether a region is triggered at note on or note off. Currently only 'attack' is supported.

21 Automation Names

Alchemy exposes numerous parameters to its host for automation, and it uses abbreviated forms for many parameter names. (These short names are consistent with the 8-character limit stipulated by the VST standard.) On this page you'll find a list of all the automatable parameters, so you can look up a full name and find its abbreviated form.

> **Note**: While most knob-based parameters in Alchemy can be automated, individual modulation depths and modulation source controls, such as LFO rate or AHDSR times, cannot. (Making these parameters directly automatable would result in a parameter list with thousands of items.) The best approach to automating one of these parameters is to assign it to a Performance Control and then automate the Performance Control.

Perform Controls

Perform Control 1 - _Contrl1
Perform Control 2 - _Contrl2
Perform Control 3 - _Contrl3
Perform Control 4 - _Contrl4
Perform Control 5 - _Contrl5
Perform Control 6 - _Contrl6
Perform Control 7 - _Contrl7
Perform Control 8 - _Contrl8
XY Pad 1 X - _XyPad1x
XY Pad 1 Y - _XyPad1y
XY Pad 2 X - _XyPad2x
XY Pad 2 Y - _XyPad2y
Attack – _EnvAtt
Decay – _EnvDec
Sustain – _EnvSust
Release – _EnvRel
Remix Pad X - _RemixX
Remix Pad Y - _RemixY

Source Controls

Source A

Amp A - S1Amp
Pan A - S1Pan
Coarse Tune A - S1TunCrs
Fine Tune A - S1TunFin
Start Time A (Position) - S1Start
Stretch A - S1Strtch
Filter Mix A - S1FilMix

Additive and VA

Additive Volume A - S1AdVol
Additive Symmetry A - S1AdSym
Additive Pitch Variance A - S1AdPVar
Additive Number of Oscillators A - S1AdNOsc
Additive Pitch Offset A - S1AdPtch
Additive Amp Offset A - S1AdAmp
Additive Pan Offset A - S1AdPan
Additive Detune Offset A - S1AdDet

Granular

Granular Volume A - S1GrVol
Granular Size A - S1GrSize
Granular Density A - S1GrDens
Granular Random Time A - S1GrRTim
Granular Random Pan A – S1GrRPan

Spectral

Spectral Volume A – S1SpVol
Spectral Low Pass Filter A – S1SpLp
Spectral High Pass Filter A – S1SpHp
Source Filters
Filter 1 Cutoff A – S1F1Cut
Filter 1 Resonance A - S1F1Res
Filter 2 Cutoff A – S1F2Cut
Filter 2 Resonance A – S1F2Res
Filter 3 Cutoff A – S1F3Cut
Filter 3 Resonance A – S1F3Res

Source B

Amp B - S2Amp
Pan B - S2Pan
Coarse Tune B - S2TunCrs
Fine Tune B - S2TunFin
Start Time B (Position) - S2Start
Stretch B - S2Strtch
Filter Mix B - S2FilMix

Additive and VA

Additive Volume B - S2AdVol
Additive Symmetry B - S2AdSym
Additive Pitch Variance B - S2AdPVar
Additive Number of Oscillators B - S2AdNOsc
Additive Pitch Offset B - S2AdPtch
Additive Amp Offset B - S2AdAmp
Additive Pan Offset B - S2AdPan
Additive Detune Offset B - S2AdDet

Granular

Granular Volume B - S2GrVol
Granular Size B - S2GrSize
Granular Density B - S2GrDens
Granular Random Time B - S2GrRTim
Granular Random Pan B – S2GrRPan

Spectral

Spectral Volume B – S2SpVol
Spectral Low Pass Filter B – S2SpLp
Spectral High Pass Filter B – S2SpHp

Source Filters

Filter 1 Cutoff B – S2F1Cut
Filter 1 Resonance B - S2F1Res
Filter 2 Cutoff B – S2F2Cut
Filter 2 Resonance B – S2F2Res
Filter 3 Cutoff B – S2F3Cut
Filter 3 Resonance B – S2F3Res

Source C

Amp C - S3Amp
Pan C - S3Pan
Coarse Tune C - S3TunCrs
Fine Tune C - S3TunFin
Start Time C (Position) - S3Start
Stretch C - S3Strtch
Filter Mix C - S3FilMix

Additive and VA

Additive Volume C - S3AdVol
Additive Symmetry C - S3AdSym
Additive Pitch Variance C - S3AdPVar
Additive Number of Oscillators C - S3AdNOsc
Additive Pitch Offset C - S3AdPtch
Additive Amp Offset C - S3AdAmp
Additive Pan Offset C - S3AdPan
Additive Detune Offset C - S3AdDet

Granular

Granular Volume C - S3GrVol
Granular Size C - S3GrSize
Granular Density C - S3GrDens
Granular Random Time C - S3GrRTim
Granular Random Pan C – S3GrRPan

Spectral

Spectral Volume C – S3SpVol
Spectral Low Pass Filter C – S3SpLp
Spectral High Pass Filter C – S3SpHp

Source Filters

Filter 1 Cutoff C – S3F1Cut
Filter 1 Resonance C - S3F1Res
Filter 2 Cutoff C – S3F2Cut
Filter 2 Resonance C – S3F2Res
Filter 3 Cutoff C – S3F3Cut
Filter 3 Resonance C – S3F3Res

Source D

Amp D - S4Amp

Pan D - S4Pan
Coarse Tune D - S4TunCrs
Fine Tune D - S4TunFin
Start Time D (Position) - S4Start
Stretch D - S4Strtch
Filter Mix D - S4FilMix

Additive and VA

Additive Volume D - S4AdVol
Additive Symmetry D - S4AdSym
Additive Pitch Variance D - S4AdPVar
Additive Number of Oscillators D - S4AdNOsc
Additive Pitch Offset D - S4AdPtch
Additive Amp Offset D - S4AdAmp
Additive Pan Offset D - S4AdPan
Additive Detune Offset D - S4AdDet

Granular

Granular Volume D - S4GrVol
Granular Size D - S4GrSize
Granular Density D - S4GrDens
Granular Random Time D - S4GrRTim
Granular Random Pan D – S4GrRPan

Spectral

Spectral Volume D – S4SpVol
Spectral Low Pass Filter D – S4SpLp
Spectral High Pass Filter D – S4SpHp

Source Filters

Filter 1 Cutoff D – S4F1Cut
Filter 1 Resonance D - S4F1Res
Filter 2 Cutoff D – S4F2Cut
Filter 2 Resonance D – S4F2Res
Filter 3 Cutoff D – S4F3Cut
Filter 3 Resonance D – S4F3Res

Morph

Morph All X - MorAllX
Morph All Y – MorAllY
Morph Time X – MorTimeX
Morph Time Y – MorTimeY
Morph Additive X – MorAddX
Morph Additive Y – MorAddY
Morph Spectral X – MorSpecX
Morph Spectral Y – MorSpecY
Morph Granular X – MorGranX
Morph Granular Y – MorGranY
Morph Filter X – MorFiltX
Morph Filter Y – MorFiltY

Main Filters

Filter 1 Cutoff – F1Cut
Filter 1 Resonance – F1Res
Filter 1 Drive – F1Drive
Filter 1 FX-Mix – F1FxMix
Filter 2 Cutoff – F2Cut
Filter 2 Resonance – F2Res
Filter 2 Drive – F2Drive
Filter 2 FX-Mix – F2FxMix
Series/Parallel Filter Mix – FSerMix

Master Amp

Master Volume – Volume
Master Amp – Amp
Master Pan – Pan
Master Coarse Tune – TuneCrs
Master Fine Tune – TuneFine
Master Portamento – PortRate

Arpeggiator

Mode – ArpMode
Latch – ArpLatch
Coarse Tune – ArpTCrs
Amp – ArpAmp
Pan – ArpPan
Split – ArpSplit
Key Velocity – ArpKVel
Octave – ArpOct
Source – ArpSrc

Effects

Acoustic Reverb

Acoustic Reverb Time – AmbTime
Acoustic Reverb Predelay – AmbPreDe
Acoustic Reverb Room Size – AmbSize
Acoustic Reverb Room Width – AmbWidth
Acoustic Reverb Diffusion – AmbDiffu
Acoustic Reverb Gate Amount – AmbGtAm
Acoustic Reverb Gate Attack – AmbGtAt
Acoustic Reverb Gate Decay – AmbGtRel
Acoustic Reverb EQ Low Frequency – AmbEqLoF
Acoustic Reverb EQ Low Gain – AmbEqLoG
Acoustic Reverb EQ High Frequency – AmbEqHiF
Acoustic Reverb EQ Hi Gain – AmbEqHiG
Acoustic Reverb Damp Low Frequency – AmbDmpLF
Acoustic Reverb Damp Low Amount – AmbDmpLA
Acoustic Reverb Damp High Frequency – AmbDmpHF
Acoustic Reverb Damp High Amount – AmbDmpHA
Acoustic Reverb Quality – AmbQual
Acoustic Reverb Variation – AmbVar

Acoustic Reverb Mix – AmbMix

Camel Reverb

Camel Reverb Pre-delay – RevDelay
Camel Reverb Damping – RevDamp
Camel Reverb Low Cut – RevLoCut
Camel Reverb High Cut – RevHiCut
Camel Reverb Size – RevSize
Camel Reverb Mix – RevMix

Delay 1

Delay L Rate - DelLRate
Delay R Rate – DelRRate
Delay L Offset – DelLOff
Delay R Offset – DelROff
Delay L Feedback – DelFback
Delay R Feedback – DelFbaR
Delay Filter 1 Cutoff - DelF1Cut
Delay Filter 1 Resonance – DelF1Res
Delay Filter 2 Cutoff – DelF2Cut
Delay Filter 2 Resonance – DelF2Res
Delay Crossover – DelCross
Delay Initial Pan - DelIPan
Delay Mix – DelMix

Delay 2

Delay 2 L Rate – DelLRat2
Delay 2 R Rate - DelRRat2
Delay 2 L Offset – DelLOff2
Delay 2 R Offset – DelROff2
Delay 2 L Feedback – DelFbac2
Delay 2 R Feedback – DelFbaR2
Delay 2 Filter 1 Cutoff – DelF1Cu2
Delay 2 Filter 1 Resonance – DelF1Re2
Delay 2 Filter 2 Cutoff – DelF2Cu2
Delay 2 Filter 2 Resonance – DelF2Re2
Delay 2 Crossover – DelCros2
Delay 2 Initial Pan – DelIPan2
Delay 2 Mix – DelMix2

Modulation FX 1

ModFX Delay – ModDelay
ModFX Depth – ModDepth
ModFX Rate – ModRate
ModFX Feedback – ModFback
ModFX Mix – ModMix
ModFX Stereo – ModSte

Modulation FX 2

ModFX 2 Delay – ModDela2
ModFX 2 Depth – ModDept2

ModFX 2 Rate – ModRate2
ModFX 2 Feedback – ModFbac2
ModFX 2 Mix – ModMix2
ModFX 2 Stereo – ModSte2

Distortion 1

Distortion Crush – DisCrush
Distortion Mech – DisMech
Distortion Tube – DisTube
Distortion Xcita – DisXcita
Distortion Post-gain – DisPGain

Distortion 2

Distortion 2 Mech – DisMech2
Distortion 2 Crush – DisCrus2
Distortion 2 Tube – DisTube2
Distortion 2 Xcita – DisXcit2
Distortion 2 Post-gain – DisPGai2

Equaliser 1

EQ Low Gain – Eq3LoGai
EQ Mid Gain – Eq3MiGai
EQ High Gain – Eq3HiGai
EQ Low Frequency – Eq3LoFrq
EQ Mid Frequency – Eq3MiFrq
EQ High Frequency – Eq3HiFrq
EQ Low Bandwidth – Eq3LoBw
EQ Mid Bandwidth – Eq3MiBw
EQ High Bandwidth – Eq3HiBw

Equaliser 2

EQ 2 Low Gain – Eq3LoGa2
EQ 2 Mid Gain – Eq3MiGa2
EQ 2 High Gain – Eq3HiGa2
EQ 2 Low Frequency – Eq3LoFr2
EQ 2 Mid Frequency – Eq3MiFr2
EQ 2 High Frequency – Eq3HiFr2
EQ 2 Low Bandwidth – Eq3LoBw2
EQ 2 Mid Bandwidth – Eq3MiBw2
EQ 2 High Bandwidth – Eq3HiBw2

Band Pass Filter 1

Bandpass Filter Low Frequency Cutoff – BpfLoCut
Bandpass Filter High Frequency Cutoff – BpfHiCut
Bandpass Filter Low Frequency Resonance – BpfLoRes
Bandpass Filter High Frequency Resonance – BpfHiRes

Band Pass Filter 2

Bandpass Filter 2 High Frequency Cutoff – BpfHiCu2
Bandpass Filter 2 Low Frequency Cutoff – BpfLoCu2

Bandpass Filter 2 Low Frequency Resonance – BpfLoRe2
Bandpass Filter 2 High Frequency Resonance – BpfHiRe2

Band Reject 1 & 2

Band Reject Mix – BpfBRMix
Band Reject 2 Mix – BpfBRMi2

Multimode Filter

Multimode Filter Cutoff – FilCut
Multimode Filter Resonance – FilRes
Multimode Filter Drive – FilDrive
Multimode Filter Mix – FilMix

Bass Enhancer

Bass Enhancer Amount – MeqAmnt
Bass Enhancer Tune – MeqTune

Compressor

Compression Amount – ComAmnt
Compression Release – ComRelse

Gain

FX Gain – AmpAmp

Pan

FX Pan – PanPan

22 Additive and spectral data

Each Source content popup menu includes a pair of commands for loading and saving additive and spectral data. The 'Save Additive/Spectral' command allows you to write a Source's additive and spectral data to either an AAZ file or a CSV file. AAZ is the compact binary format that Alchemy uses by default for this data. CSV, or Comma Separated Values, is a human-readable format that can be edited in spreadsheet software (such as Microsoft Excel) or in a text editor (such as Notepad or TextEdit).

The 'Load Additive/Spectral' command reads existing additive and spectral data from an AAZ or CSV file and loads it into a Source; this command also configures the Source properly to play the newly loaded data. For example, if you load a file that contains additive data, the Source's Additive element will be toggled on and set to Additive mode, and the NOsc parameter will be set to an appropriate value based on the number of partials required by the data.

22.1 Creating your own Additive/Spectral Sounds in CSV Format

If you want to create your own additive and/or spectral sounds outside Alchemy and import them, then the easiest approach is to create a 'template' by first exporting a CSV file from Alchemy. To begin, open Alchemy and import a sample into Source A using either additive, spectral, or additive+spectral analysis, depending on which of the synthesis methods you wish to use in your own sounds. Then select 'Save Additive/Spectral' from the Source A content popup menu. A 'Save File' dialog will open. Select a location in which to save the file and enter a filename **ending with the extension '.csv'**. If there is more than one zone in the selected source then one CSV file will be created for each zone.

> **Note**: The '.csv' extension is important. If you use '.aaz' instead, or if you use no extension on the filename, then the result will be an AAZ file rather than a CSV file.

If you now open the file that you just saved using your spreadsheet or text editing software then you will see the following arrangement of fields and headers. Your spreadsheet software may ask what type of delimiters are used in the CSV file; if this happens you should specify that it is 'tab-delimited'.

```
AAZ
AttackPeakTime  0.02467120
LoopStart  1.38038552
LoopEnd  4.98585033
Length  6.52133799
Additive
PitchOffset
Time  0.00414966  0.00829932  0.01244898 ...
PitchDev  -0.03084946  -0.03084946  -0.03084946 ...
NumPartials  122
StartPhase1  0.26740721
Time1  0.00311110  0.00321110  0.00561085 ...
Amp1  0.00000000  0.42123079  0.65871763 ...
Pitch1  0.02609122  0.02609122  -0.00004935 ...
Pan1  0.50000000  0.49999982  0.50000000 ...
...
...
...
Spectral
NumBands  257
NumKeyframes  1
KeyframeTimes  0.00000000
KeyframePhase1  0.00000000
KeyframePhase2  0.00000000
KeyframePhase3  0.00000000
...
...
...
Amp1  16.37264061  25.68283081  21.17712212 ...
Freq1  0.00000000  0.00000000  0.00000000 ...
Amp2  9.53631592  17.12357521  11.94564819 ...
Freq2  100.00543976  68.44330597  32.56330109 ...
Amp3  12.63044643  5.10157347  2.92674470 ...
Freq3  274.27560425  55.12643433  139.73352051
```

22.2 Data Format Description

The following table describes all the Fields/Headers, listed in the order they are expected to occur.

Field/Header	Description
AAZ	File header. Every CSV file containing additive and/or spectral data must have this, and only this, in its first line.
AttackPeakTime	The time in the sound at which the peak of the attack stage occurs. If the sound has no attack stage then this should be zero. The units for this field are seconds.
LoopStart	The time in the sound at which the first loop marker should be placed. The units for this field are seconds.
LoopEnd	The time in the sound at which the last loop marker should be placed. The units for this field are seconds.
Length	The total length of the sound. This is where the last warp marker should be placed. The additive/spectral data should have no points after this time and should fall to zero before they reach the end.
Additive	The header for the additive data. If there is no additive data then you should leave out this header along with all of the additive fields.
Pitchoffset	The header for the additive pitchoffset envelope data. The pitchoffset envelope describes the fundamental pitch envelope. In other words, it is an envelope that shows how the pitch of the whole sound changes over time.
Time	The times at which the points on the pitchoffset envelope occur in units of seconds.
PitchDev	The pitch deviation of each of the points in the pitchoffset envelope expressed in semitones. A value of zero means that the fundamental frequency is not altered. There must be the same number of PitchDev values as there are Time values.
NumPartials	The number of additive partials. This must not be greater than 600.
StartPhase<N>	The phase of the Nth partial at the start of the sound. Its value should be in the range 0–1, with 1 representing a phase shift of a full cycle. The <N> should be replaced by the number of the current partial, starting with 1.
Time<N>	The times at which the points in the amplitude, pitch and pan envelope for the Nth partial occur in units of seconds. The <N> should be replaced by the number of the current partial, starting with 1.
Amp<N>	The amplitudes at each point in the Nth partial's envelope. The amplitude values should not be greater than 2. The <N> should be replaced by the number of the current partial, starting with 1. *See the note below for the relationship between the linear amplitude values used in the CSV format and the dB values displayed in the Alchemy GUI.*
Pitch<N>	The pitch deviations at each point in the Nth partial's envelope. *The format currently in use for these values has been judged excessively complex, and a change to a simpler scaling system is tentatively planned for version 1.13. Therefore users are advised to wait for the new system to be put in place before*

	undertaking projects that manipulate partial pitch data in CSV files.
Pan<N>	The stereo position at each point in the Nth partial's envelope. The range of the values is 0–1 where 0 represents far-left, 0.5 is centred., and 1 is far-right. The <N> should be replaced by the number of the current partial, starting with 1.
Spectral	The header for the spectral data. If there is no spectral data then you should leave out this header along with all of the spectral fields.
NumBands	The number of spectral bands. This number must be either 129, 257, 513, or 1025.
NumKeyframes	The number of phase keyframes. These are used to reset the phase of the partials at particular points in the sound. When Alchemy analyses sounds it looks for transients and places these phase keyframes just before them to ensure that they sound as punchy as in the original sound.
KeyframeTimes	A list of the times at which the phase keyframes occur. There should be the same number of times as given in the NumKeyframes field.
KeyframePhase<N>	A list of the phases for each of the bands at the Nth keyframe. There should be the same number of phases given here as given in the NumBands field. The values should be in the range 0–1, with 1 representing a phase shift of a full cycle. The <N> should be replaced by the number of the current keyframe, starting with 1 and ending with the number of keyframes given in the NumKeyframes field.
Amp<N>	A list of the amplitudes for each for each of the bands at the Nth time step. The length of a time step is constant for a given value of NumBands and can be calculated by the formula NumBands / 88200.0 in seconds. The <N> should be replaced by the number of the current time step, starting with 1 and ending with the number of time steps. *See the note below for the relationship between the linear amplitude values used in the CSV format and the effective dB values.*
Freq<N>	A list of the absolute frequencies for each of the bands at the Nth time step (see above), expressed in Hz. Alchemy expects the frequencies in each band to be harmonically related with a limited variation and the frequency of the first band being centred. on 0Hz (DC). The maximum frequency variation for any band is 22050.0 / NumBands.

22.3 Amplitude scaling systems

The partial amplitudes reported numerically in the Alchemy GUI use a dB scale, and the height of each corresponding partial bar is the cube root of the dB value (in order to ensure that meaningful differences are clearly visible). CSV files use Alchemy's internal, linear amplitude scale. Conversion between the linear and dB scales is governed by the standard equations illustrated to the right.

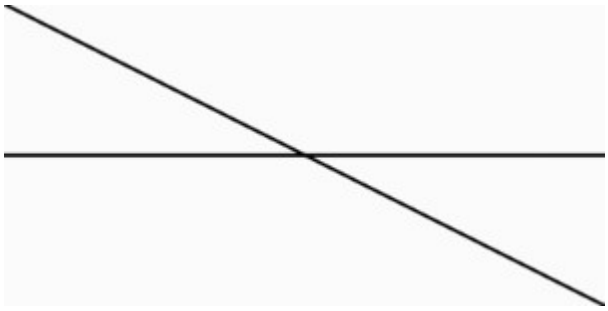
$$\mathrm{LINVAL} = 10^{\,\mathrm{DBVAL}/20}$$

$$\mathrm{DBVAL} = 20 \times \log_{10} \mathrm{LINVAL}$$

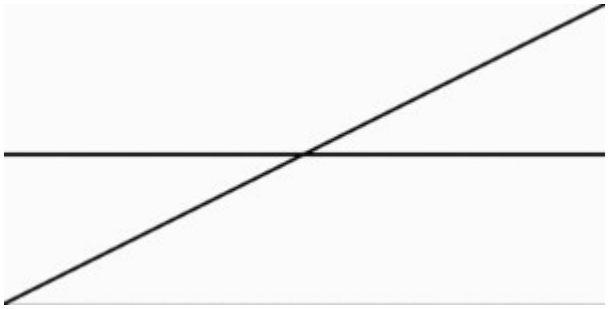
23 LFO Shapes illustrated

23.1 Basic

RampDown

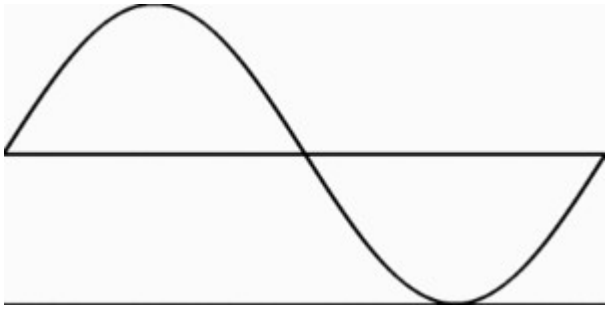


RampUp

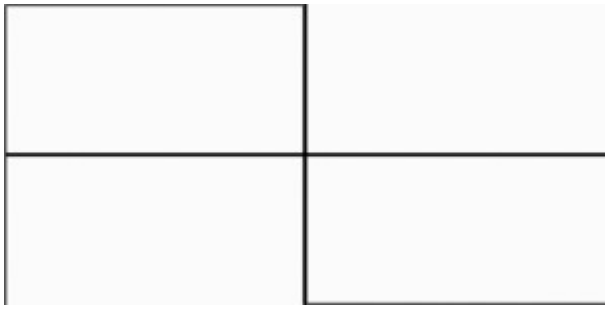


RandGlide and RandHold are described on the main LFO page.

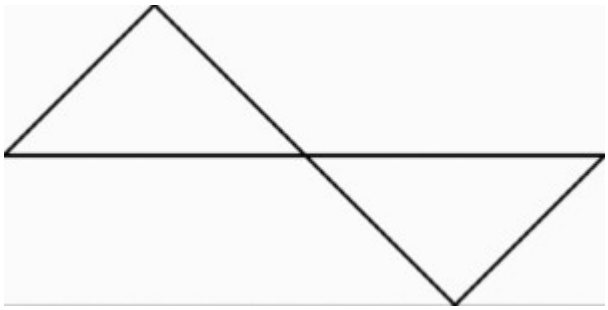
Sine



Square

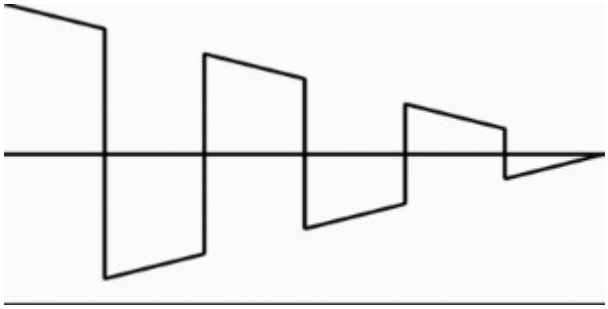


Triangle

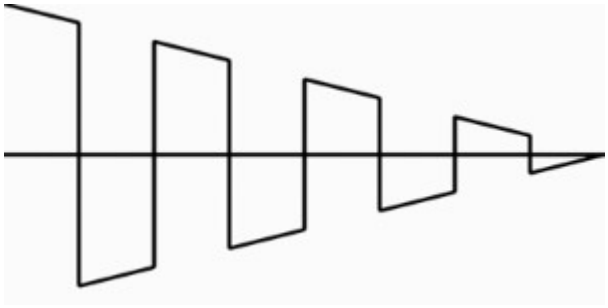


23.2 Serial-Angular

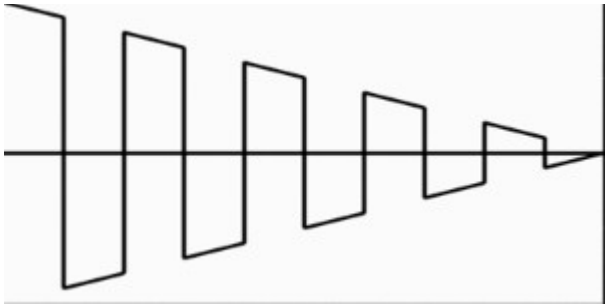
For-Sqr03



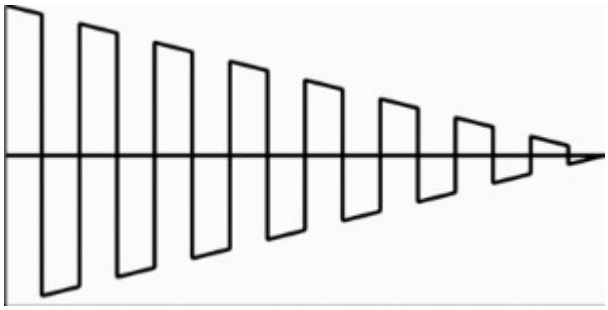
For-Sqr04



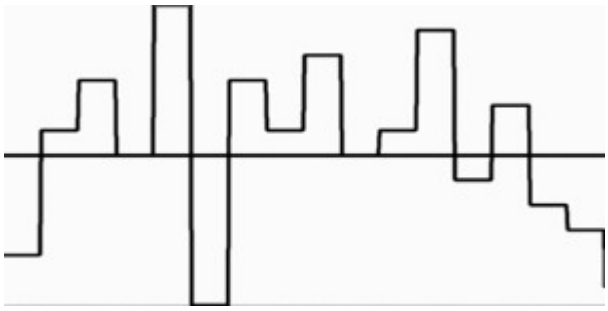
For-Sqr05



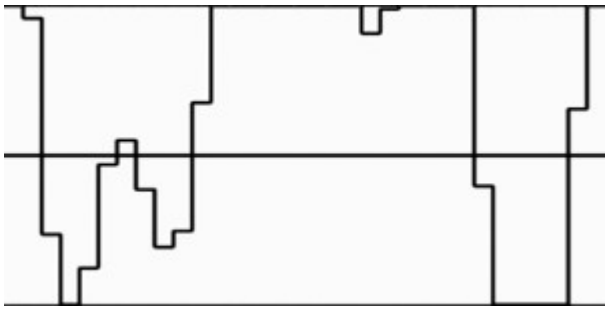
For-Sqr08



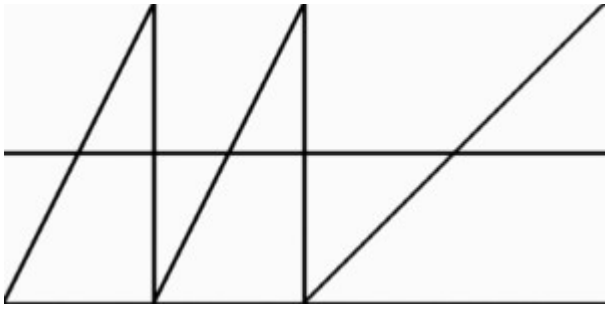
Rnd-16Step



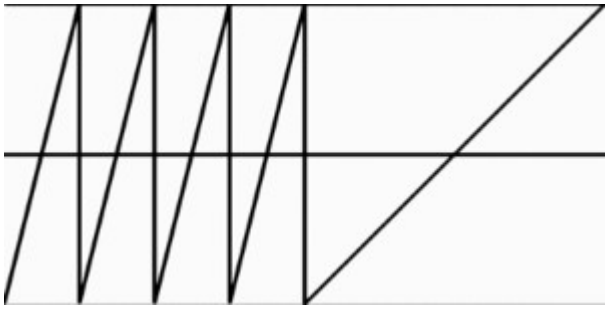
Rnd-32Step



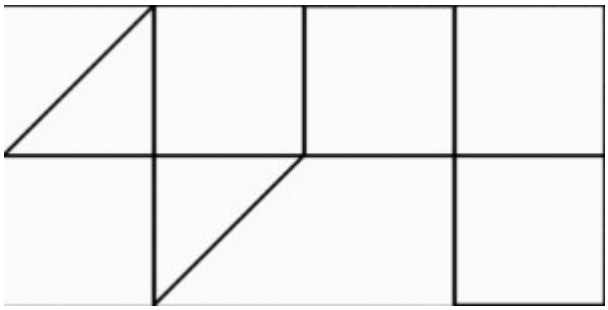
Ser-Saw2



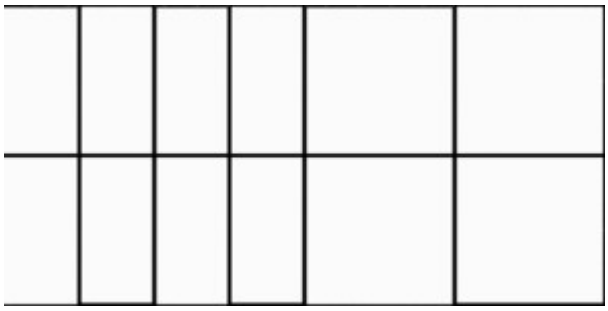
Ser-Saw4



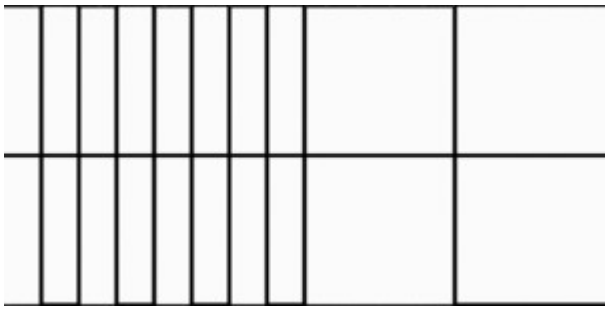
Ser-SawSq



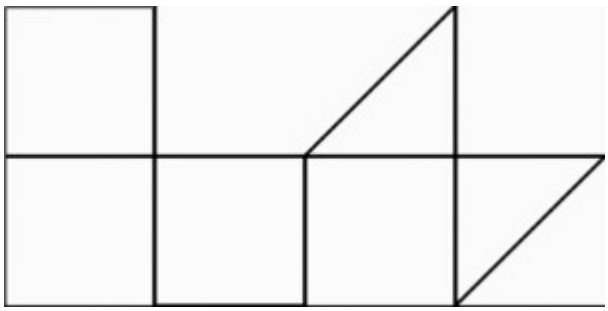
Ser-Sqr2



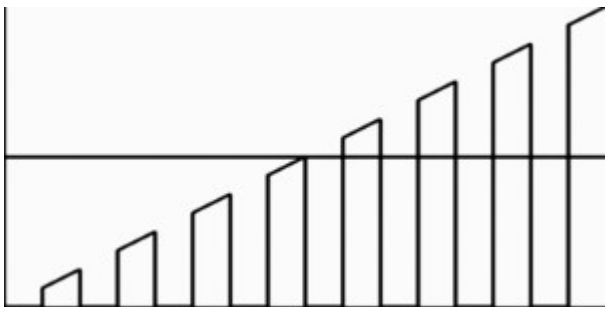
Ser-Sqr4



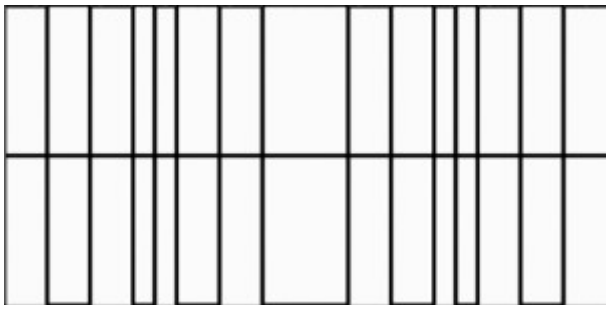
Ser-SqrSaw



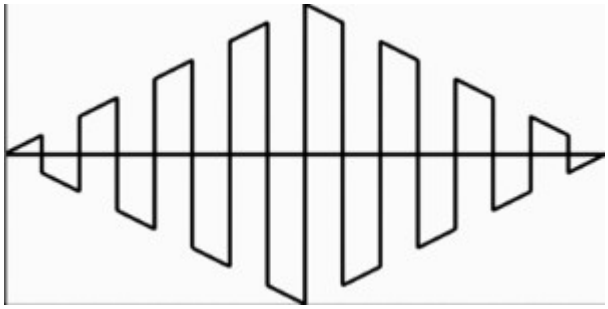
Ser-8RampUp



Ser-AM0702

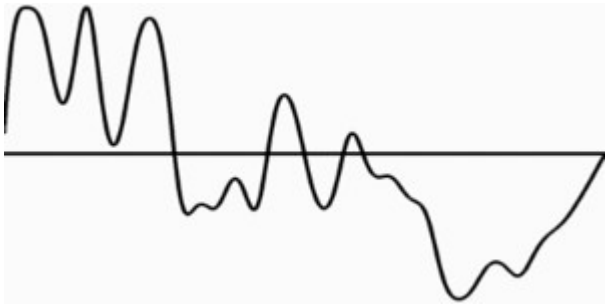


Ser-AM0Tri

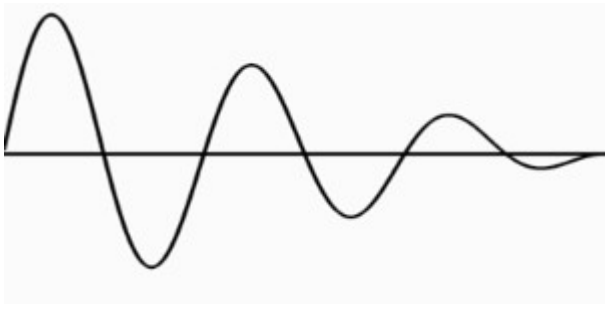


23.3 Serial-Smooth

Asy-023

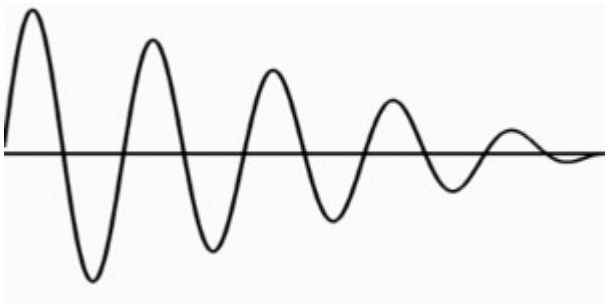


For-03

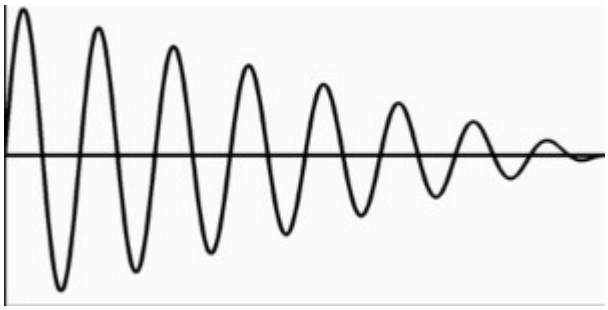


For-04

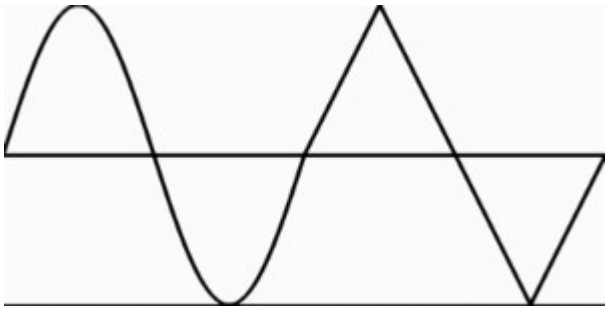
For-05



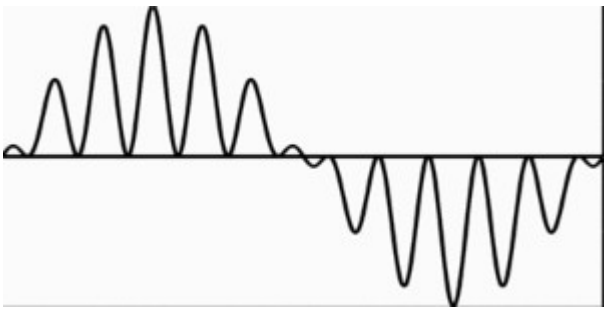
For-08



Ser-SinTri

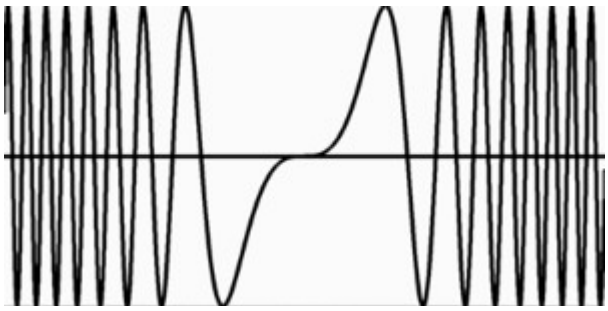


Sin-AM0112



Sin-AM0803

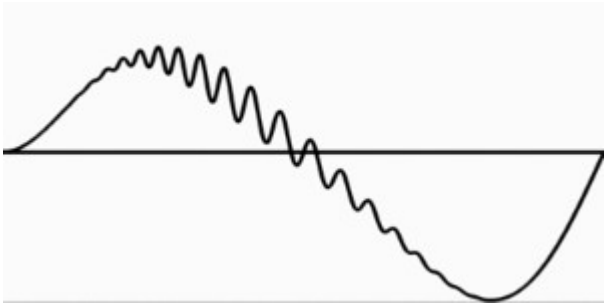
Sin-FM16



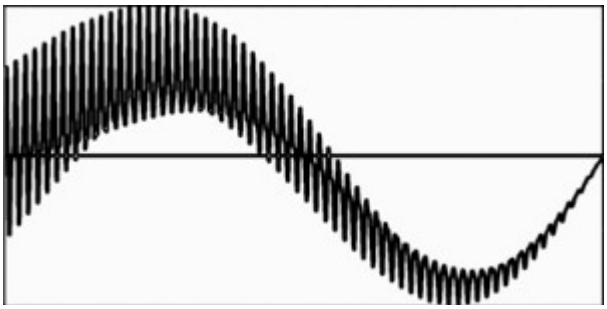
Tri-AM0501

23.4 UHF

Asy-037

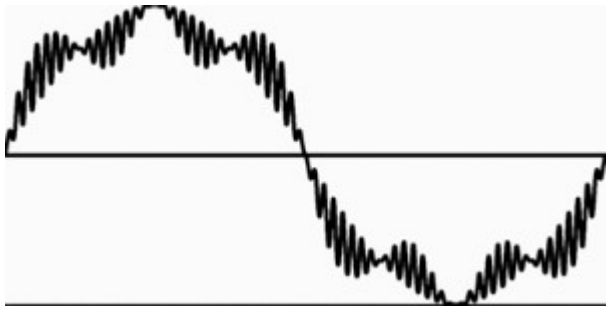


Asy-042

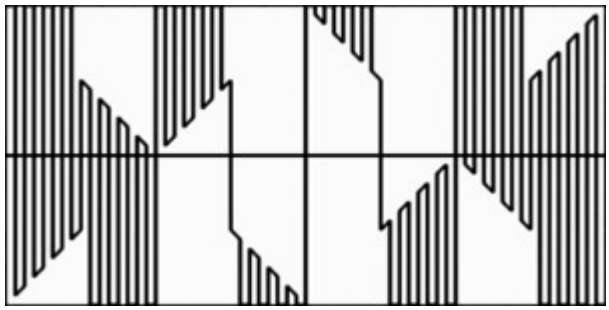


23.4 UHF

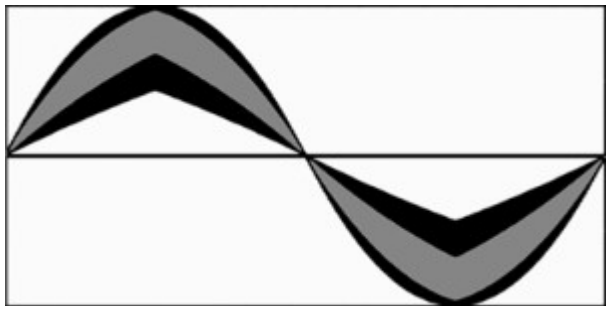
Sin-AMFM2



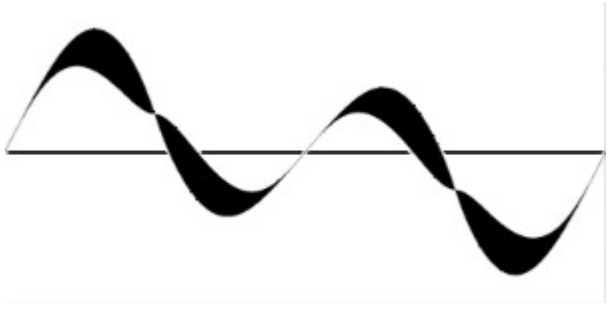
UHF-AMNS1



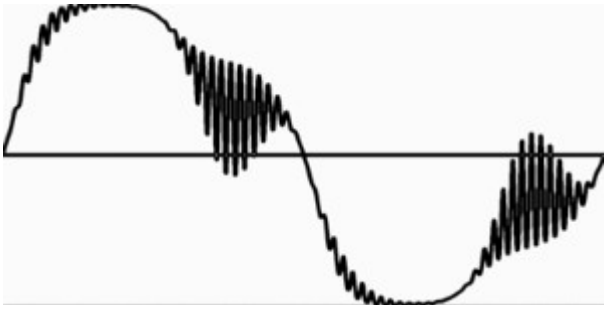
UHF-Crawlr



UHF-DNA

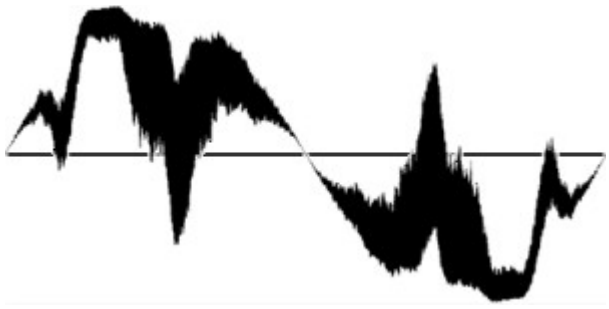


UHF-Salman

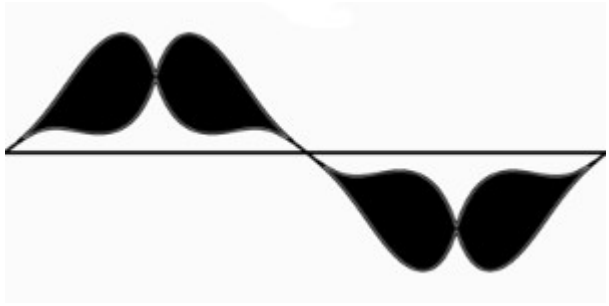


23.4 UHF

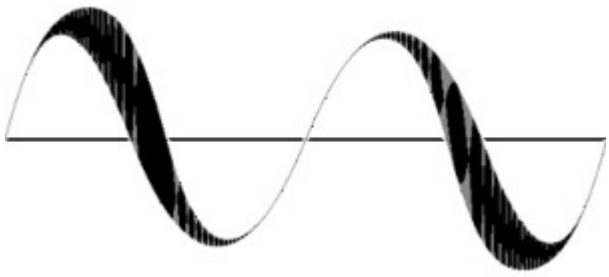
UHF-Sieral



UHF-Tennis



UHF-Trump



24 Custom oscillators and LFOs

You can extend the functionality of Alchemy by creating and installing custom waveforms to load in Alchemy's oscillators and LFOs.

> **Note**: This is an advanced technique that requires audio editing software with specialised features. In many cases it makes better sense to create new oscillator waveforms using Alchemy's built-in additive editor, although these results require Alchemy's additive engine for playback and cannot be loaded into a single oscillator for VA applications. Similarly, in many cases it makes better sense to create new LFO shapes using a looped MSEG. (Or, if you need a custom shape plus the ability to modulate the rate at which it cycles, you might use a Ramp Up LFO and apply a ModMap to reshape it as needed.)

24.1 Custom LFO waveforms

An LFO waveform is defined by a RAW file consisting of 2048 contiguous little-endian 32-bit floating point (single-precision) numbers in the range from –1 to 1, representing a single cycle of the waveform. The resulting file should be placed in the following sub-folder:

Alchemy/Libraries/WaveLfo

24.2 Custom Oscillator waveforms

An oscillator waveform is defined by a RAW file containing 20 cycles of the waveform, each lowpass filtered with a progressively lower cutoff frequency. This is to allow the oscillator to be used over a wide range of frequencies without significant aliasing. The cutoff starts at one half-octave below the Nyquist frequency and then goes down in half-octave decrements; in other words, the frequency is divided by the square root of 2 each time. For more experimental results it would be possible to place whatever waveform you like into each slot, allowing for the use of different waveforms at different pitches. The resulting file should be placed in the following sub-folder:

Alchemy/Libraries/WaveOsc

A possible workflow for constructing these oscillator waveform files begins with 2048 samples representing the full bandwidth version of the waveform. Apply the necessary filtering — an FIR filter is suitable for this task — and append the filtered result to the original waveform. Repeat this process 19 times while progressively increasing the amount of filtering according to the specifications above.

25 Cameleon to Alchemy

While many of Alchemy's features are new, others are inherited from Camel Audio's Cameleon 5000 synthesiser. This gives Alchemy the ability to replicate the capabilities of its older sibling. If you are coming to Alchemy from Cameleon, this page will help you understand how key features in the two instruments correspond.

Voices A–D in Cameleon correspond to ***Sources A–D*** in Alchemy.

The **Harmonic** component of a Cameleon voice is equivalent to the ***Additive*** element of an Alchemy source.

The **Noise** component of a Cameleon voice is equivalent to the ***Spectral*** element of an Alchemy source. (More specifically, set the Spectral element to NOISE-RESYNTH mode.)

While voices in Cameleon are invariably combined by **morphing**, Alchemy offers both ***crossfading and morphing*** capabilities. The difference between a crossfade and a morph is discussed on the Morph page of this manual. To replicate Cameleon's behaviour, stick with morphing.

While the **'Morph Square'** is a fixed part of the Cameleon interface, Alchemy offers two ***freely assignable XY Pads*** in its Perform section that may be configured to control morphing. For details, see Where did the 'Morph Square' go? and Basic additive and spectral morphing on the Morph page of this manual.

Similarly, Alchemy can be configured to replicate Cameleon's **Amplitude, Harmonics, and Noise** morph modes (see Morphing different aspects of the sound) and its Morph Timeline (see MSEG-driven morphs).

Increasing Harmonise in a Cameleon voice ('Harm' knob in the Harmonics section) is equivalent to ***decreasing Pitch Variation*** in an Alchemy source ('PVar' knob on the ADDITIVE sub-page).

The **Formant Filter** on Cameleon's Effects page corresponds to the ***Formant ModMap*** feature in each Alchemy source, found in the Amp menu on the ADDITIVE subpage. For details, see Using the Additive Formant Filter. Note that when Alchemy is morphing (rather than crossfading) sources, the Amp menu selection for Source A applies to all four sources.

The multi-mode **Filter** on Cameleon's Effects page is applied separately to each polyphonic voice; then the voices are mixed and the result is passed through the remaining effects blocks (Distortion, Chorus, Stereo Delay, Reverb). To reproduce this scheme faithfully in Alchemy, use one of the Main Filters — in one of the six 'BQ' or 'Fat' modes listed at the top of the filter type menu — and use the same effects blocks in the same order as you would in Cameleon. However, there is a much larger range of filtering and effects processing possibilities to be explored in Alchemy; see the Filter and Effects pages of this manual for details.

Both Cameleon and Alchemy provide a RANDOM button in their title bars, to intelligently generate new random presets. More localised randomisation is available in certain Cameleon modules (via a distinctive **'Camel' button**) and in many Alchemy modules (via a ***Randomise*** command in the Source content menus and various FILE menus).

25.1 Preset types and terminology

Both Cameleon and Alchemy allow you to load and save whole presets (instrument settings) as well as a variety of smaller component settings. The table below lists these from large to small, showing correspondences between the preset types of the two instruments.

Cameleon	**Alchemy**
Instrument (.c5i)	Preset (.acp) — *see note below*
Voice Program (.c5v)	Source (.src)
Harmonics (.c5h) and Noise (.c5n)	Additive/Spectral data (.aaz binary file, .csv text file)
Morph Envelope (.c5m) and Formant Filter (.c5f)	MSEG (.mse) and ModMap (.mma)
–	LFO (.lfo), AHDSR (.ahd), Sequencer (.seq), ARP (.arp)
–	Acoustic/Camel Reverb, ModFX (.acr)
–	Delay (.dly)

> **Note**: .c5i files that originated in Cameleon can be translated into Alchemy presets (.acp files) via the FILE > Load command described on the Title bar page of this manual.

26 Troubleshooting

If Alchemy is not behaving the way you expect, here are some things to try.

26.1.1 There is distortion — how do I get rid of it?

First, try turning down the main Volume control (in the Title Bar), as the sound may be clipping when it enters your sequencer/host. If this does not solve the problem, then turn down all the Source Amp controls by equal amounts (a reduction of 6 dB or more may help). Sometimes by soloing different Sources and turning different elements on and off, you can isolate the problem in an individual source or element and resolve it by reducing the Source Amp or element Vol control. If distortion remains, it is likely to be caused by a deliberate distortion effect. Check to see if a Source or Main Filter is set to the 'Tube', 'Mech', or 'Ring' type, or if a conventional filter type has a high Drive setting, or if the Effects rack contains a Distortion module.

26.1.2 The sound is too quiet — how do I make it louder?

Turn up the Master Volume and Master Amp controls (to their maximum levels if needed). If it is still too quiet, turn up Source Amp controls and individual element Vol controls. If you still don’t have enough volume, try increasing the Drive setting of any conventional filters you are already using, or set an unused (source or main) filter to the 'Mech' type and boost its Resonance knob enough to raise the gain without causing noticeable distortion. You can also add a Compressor module to the Effects Rack.

26.1.3 There is a 'click' at the start of each note — How do I get rid of it?

You have probably set the attack time of the Master Amp envelope to zero, which means the sound reaches full volume instantly. Try increasing the Attack time very slightly. A setting of 0.001 s, which Alchemy uses by default, often works well if you want a fast but click-free attack.

26.1.4 There are 'stuck' notes — how do I eliminate them?

If Alchemy continues to play notes after you release them, the likely culprit is the modulation routing of the Master Amp parameter. By default, AHDSR 1 modulates this parameter, acting as a 'master amp envelope'; and whenever an AHDSR module is routed to the Master Amp parameter, the modulation depth is locked at 100%, which ensures that the amplitude will fall to zero at the end of the release stage. To resolve a 'stuck' notes issue, try the following:

26.1.5 When using an MSEG as the 'master amp envelope', ensure that the modulation depth is set to exactly 100%.

If you have not assigned either an AHDSR or an MSEG to modulate the Master Amp, add an AHDSR in this role. By setting an Attack time of 0 sec and a Sustain level of 100%, you can preserve the existing amplitude characteristics while eliminating the 'stuck' notes.

Another possibility, external to Alchemy, is that your sequencer may contain MIDI data with a missing 'Note Off' message.

26.1.6 Alchemy's CPU usage is too high - how do I reduce it?

First, if you are running your host with less than 256 samples (or buffers) of latency, we recommend increasing it to 256 samples. Beyond that, there are a variety of strategies. The best solution depends on the details of your preset, so try the following, checking to see on one hand how much the CPU load is reduced and on the other hand how much the sound changes.

If using the Acoustic Reverb module with Quality set higher than 40%, reduce this setting to 40%. Or consider switching to the Camel Reverb effect.

Reduce polyphony. (Your needs for a particular preset may vary, but the following minimum voice counts were established for the factory presets: Keys, Organs 8; Pads, Brass, Guitars, Mallets, Strings, Synths, Vocals 6; Soundscapes, Drums, Arpeggiated 3; Bass, Loops, Sound Effects, Leads, Woodwinds 2. Of course some presets are monophonic by design.)

Turn STEREO off for all Sources which use the Additive element or the Source filters.

If using a HQ filter type, switch to the corresponding non-HQ type. (For example, switch from 'LP2-SVF HQ' to 'LP2-SVF'; see the Filter page for details.)

Try switching off Source filters if their contribution to the preset is not significant.

If using the Additive element (in ADD or VA mode), use fewer oscillators. Listen to each Source as you reduce its NOsc control to ensure that you preserve the essential sound. Alchemy processes oscillators in blocks of four, so NOsc settings that are multiples of four (4, 8, 12, 16, …) give you the most oscillators for the least CPU cost.

The Additive element in ADD mode benefits from special optimisations, cutting CPU load in half when all of the following conditions are met:

Oscillator wave is Basic > Sine.

Symmetry is set to exactly 50% and not modulated.

No modulation is applied to the Master Coarse and Fine Tune, Source Coarse and Fine Tune, and Additive element Pitch controls.

If using a Granular element with grain Density set above 4, reduce this setting to 4.

If the sound has a long release time, try reducing it; it may be more efficient to achieve similar effects with Delay and/or Reverb effects.

If using the Granular element with Stretch fixed at 100% and with no special granular manipulations, try switching the element to Sampler mode.

Use fewer effects.

Use fewer sources.

If in Xfade mode, use an analogous Morph mode instead. This makes a large difference for Additive and Spectral elements (although it may significantly change the sound of the preset).

26.1.7 A dialog box has appeared saying 'Please select a location and filename for supporting data files...' — what should I do?

This dialog box appears when your sequencer is saving a song that contains more than 1MB of additive or spectral analysis data. You should click ‘OK’. In the following File Save dialog you should navigate to the folder in which your song is saved and enter a suitable name for the Alchemy preset file that will be saved there. This file will be referred to by the song so it is a good idea to keep it in the same folder.

The reason why it is necessary to save this data outside the song is because some sequencers do not allow more than 1MB of data to be stored inside a song by a plugin. Known examples of this are Logic and GarageBand. To find out if your sequencer has this limitation, please contact the sequencer manufacturer. If your sequencer does allow you to save more than 1MB of plugin data within a song, then you can increase this threshold by changing the value of the MaxChunkSize parameter within the AlchemyConfig.txt file that is located in your plugin folder. The MaxChunkSize parameter specifies the threshold in bytes. Increasing this value when your sequencer does not support it will lead to a crash when saving songs.

26.1.8 There are no presets — why is that?

You should be able to browse the Factory presets, as well as any User presets you have saved, and any add-on banks of presets you have installed, via the Bank, Category, and Preset fields in Alchemy’s Title bar. If the expected content is not displayed when you click these fields, then you may not have installed the Bank folders where Alchemy expects to find them. Locate these folders on your computer and move them to the Alchemy data folder (Windows) or /Library/Application Support/Camel Audio/Alchemy/Presets (Mac); Alchemy will find them the next time it is loaded.

If you cannot find Alchemy’s Factory bank anywhere on your computer, go to the Camel Audio web site at http://www.camelaudio.com, and log into your user account by clicking on the 'Log in / create user account' link at the top right-hand corner of the page and entering your email address and password. Then click on the ‘Downloads’ link in the Support Menu on the left of the page, and download Alchemy Factory Presets (a .CamelSounds file that will be expanded by the installer) to your desktop. Double-click the Camel Sounds file or run the Add-On Installer, found in the Start menu under Programs->Camel Audio->Alchemy (Windows) or in /Applications/Alchemy (Mac).

26.1.9 Alchemy is asking for a 'keyfile' — what should I do?

The keyfile is your license to use Alchemy, and downloading it is a required step in the installation process. You can download the keyfile for your purchase (of Alchemy, or of an add-on bank for it) by logging in to your account at http://www.camelaudio.com and going to the Downloads page; see the Installation and requirements page of this manual for details.

If Alchemy or its add-on content is not already authorised, it looks for the keyfile on your desktop (and moves it automatically to the proper location). Please ensure you have placed your keyfile on the desktop, and then restart

your sequencer. If you continue to see the keyfile message, please contact customer service for assistance.

27 Sound-designer Guidelines

Because of its flexibility and power, Alchemy is open to a variety of approaches to preset design. However, Camel Audio recommends certain 'best practices' for creating presets, which are summarised on this page. Think of this as the Camel Audio **house style** for preset design. **You should follow these guidelines if you are designing commercial presets for Camel Audio**, and you may find them useful if you are a third-party preset developer, too.

In general terms, Camel Audio is interested in musically useful, playable sounds rather than entire songs in one preset or incredibly odd sounds. Having said that, remix pad variations can be more unusual. We are keen on having new sounds, especially ones that use features unique to Alchemy — we don't just want to recreate classic sounds. We'd like sounds that are suitable for use across many genres, including current dance music and film and TV soundtracks.

You are free to use sample content from the Alchemy library, or self-created samples. If you do decide to supply us with new sample content (whether in .wav or .aaz format — from a legal standpoint it's the same), we will ask you to sign a contract explaining that you take legal responsibility for declaring that the samples are your own work and do not in any way use any copyright-protected material. (You may not sample any copyright-protected material, which includes all sample libraries. You can sample virtual and real instruments if the product does not use samples and you don't sample existing presets.) The contract will mean that if someone were to attempt to sue for damages due to violations of copyright, you would be legally liable. Please avoid risks!

If you'd like your presets to be considered for release by Camel Audio (and we're always looking for talented designers who make cool sounds) we strongly urge you to send us your first few completed patches before making hundreds, so we can provide you with technical feedback and advice, and point out any potential problems. While we've tried our best to make the guidelines as clear and concise as possible, we know from experience that it's not easy to anticipate every eventuality, and sending a sample of your preset work early on can save everyone a lot of hassle and unnecessary work.

27.1 Preset Design Criteria

Every preset you submit to Camel audio should adhere to the following guidelines.

The presets in the existing Factory banks (main and add-ons) satisfy these criteria, so you can check them out to get an idea of what is required. But you should still carefully read these guidelines, of course!

Keep preset names short and meaningful

The names should be no more than 28 characters long and ideally shorter. Names should use only upper and lower case letters, plus spaces and hyphens. Each word should start with an upper case letter, although short linking words such as 'of' and 'the' should remain in lower case. Preset names should not include any protected company or brand names. Instead, choose a name which is similar, so that knowledgeable users will recognise the reference - e.g. Kog instead of Korg, or Yuno instead of Juno.

Try and give each preset a unique and descriptive name that suits the sound or is evocative of its character or intended use, rather than calling it something generic or cryptic like 'BassGuitar03' or 'Additive Pad'. No sound-designer names or initials, and no category tags, should be inserted in the preset name. (Alchemy displays a sound-designer credit in the main parameter value display when each preset is loaded, and categories are managed as part of the preset folder structure.)

Patches should be the right volume

Presets that are too loud will clip (internally, or in the host), while presets that are too soft won't balance well in a mix. You should aim to make all patch and variation levels balanced, so there are no drastic changes in volume switching from preset to preset, or from variation to variation. You should aim for a peak level of –2db for each snapshot in every preset.

> **Note**: The dedicated 'Snapshot Vol' knob in the Perform section gives you control over the level of each snapshot. (Presets created for the earliest versions of Alchemy tend to use Performance Control 8 for this purpose, since the Snapshot Vol feature was added in a later update to Alchemy.) Sometimes when dragging the remix pad, the movement of parameters may cause louder volumes than any individual snapshot has stored. This is acceptable, so long as each variation is set to peak at –2db.

For patches with a polyphony of 6 notes or more, please trigger a 6 note chord at maximum velocity (127) and set levels to peak at -2db. Please trigger several different chords across the keyboard to confirm your levels. For patches

with lower polyphony, please use the same method, but with a smaller chord or a single note — the number of notes you trigger should match the value of the Voices parameter.

Make sure all perform controls are assigned and conform to the guidelines

Ensure that all perform controls are assigned to something and have suitable names. Control names should be no more than 10 characters long, and each word should start with an upper-case letter.

The Auto-Assign feature often gives the best starting point; but creativity — including the assignment of knobs to multiple parameters and the use of ModMaps for custom scaling — is encouraged.

Please follow convention where possible when making assignments. For instance, an FX Mix knob should increase the 'wet' FX signal as it is turned up, and a filter cutoff should close the filter as it is turned down. This is obviously not *always* desirable (e.g. if you have multiple parameters assigned to a single knob) but in general you should follow conventions so that the behaviour of each control is intuitive for users.

There is a **standard Perform layout**, and sound designers are urged to follow it. Occasionally, a preset design is driven by some concept that requires special performance controls. (For instance, if you create a preset that distributes four different drum sounds across the keyboard, it might be important to designate two knobs per drum sound for tone controls.) But in the vast majority of cases, the standard layout should be followed as closely as possible:

Knobs 1–2 are for timbre, pitch, other Source parameters. There is a degree of flexibility with these two controls, but if there are assignments for Source parameters, this is the primary place they should be.

Knobs 3–4 are for filter cutoff and resonance. This pair of controls should be the primary place users go to tweak filtering. If your preset design doesn’t involve a filter, consider adding one in its fully open state in the Effects stage; then users get your intended design plus the option to apply filtering.

Knobs 5–6 are for effects levels or other effects parameters. It’s always good to allow users to tailor the effects levels to suit their needs.

Knobs 7–8 are for controlling a preset’s rhythm or movement. There is a degree of flexibility with these two controls, but they are often assigned to LFO or Arpeggiator parameters, or to anything that causes or influences movement in the sound.

XY square 1 is for morphing among Sources. If more than one Source is used, this XY control is the place to assign control of morphing or crossfading. (If not, the assignment of this square is flexible.)

XY square 2 is flexible. It is often used for control over additional effects parameters, or for features that are unique to a particular preset.

The Attack/Decay/Sustain/Release knobs are master envelope controls. They should normally control the full range of the corresponding parameters in AHDSR 1. (The ‘Auto Assign Empty’ feature can take care of this for you.) If you are using an MSEG rather than an AHDSR in the role of master envelope, these controls may be left unassigned.

The **ModWheel** should be linked to the most playable performance control. The Remix Pad key-switching **Octave** should be set to ‘Off’, so that this feature doesn’t interfere with normal playing across the entire keyboard. And on the Arp page, the arpeggiator’s **Latch** mode should be set to **Off**.

Set useful modulation ranges in all perform controls

Optimally, all performance controls should be restricted to their most useful minimum and maximum modulation ranges. Special attention needs to be paid to mod wheel assignments so that settings of 50% or less cannot cause undesirable results, such as a dropout of the sound from too low a minimum low pass cutoff setting. This is particularly important for those who use controllers with spring loaded joysticks.

Make sure the perform remix variations are assigned and of the right volume

All eight remix snapshots should be assigned with different settings. All settings should be of approximately the same volume, in line with the volume guidelines above. Once again ‘Auto Assign’ is a good starting point, but be sure to check the results and make any necessary adjustments.

Preset categorization

Create a folder with your name (or the desired bank name) in Alchemy/Presets; inside the new bank folder, create category folders for each of the standard categories used in your bank. The standard categories used by Camel Audio are: Bass, Brass, Drums, Guitars, Keys, Loops, Mallets, Organs, Sound Effects, Soundscapes, Strings, Synth

Leads, Synth Pads, Vocals, and Woodwinds. These folder names are likely to be used in future versions of Alchemy for rating and choosing presets, so you should stick with them. Your bank may use all of these categories, or only a subset of them.

While the meaning of each category is not set in stone, here are a few guidelines to help you choose the most appropriate folder for each of your presets.

Bass: all instruments primarily intended for use in lower registers. These should have their Master Course Tune parameter set to a value such as –12 or –24 so that the 'best' range is in the middle of the keyboard.

Drums: mapped drum kits or individual drum/percussion sounds.

Keys: instruments played via a keyboard, such as acoustic or electric pianos, harpsichords, clav. (Organ is a separate category.)

Mallets: struck instruments typically made of glass or metal, such as bells, vibes and xylophone.

Pads: musical sound with well defined pitch, suitable for playing sustained chords in a supporting role with 6-note or greater polyphony.

Rhythmic: all presets that use Alchemy's Arpeggiator module for their default sound, or repetitive sounds that don't rely on the ARP, including drum grooves, riffs, and other sounds containing sequenced elements.

Sound Effects: non-melodic or noise-based sounds, either one-shot or looped.

Soundscapes: usually with more movement and greater evolution over time than pads, and generally used as a backdrop. They are not necessarily tuned to a scale, but usually are more musical and tonal than Sound Effects. Mostly used with 4-note or lower polyphony.

Samples

When you save a preset with additive and/or spectral data, this data (in the form of an .aaz file) is automatically saved alongside the preset (.acp file) itself. But when you save a preset that references wav/aiff/sfz files — the data types used for GRANULAR/SAMPLER playback — saving your preset creates a reference to the sample data in its original location. For factory samples, this works fine, because users will have the same samples in an equivalent location on their own machines. But for your own custom sample data, you'll need to submit copies of the samples to be distributed along with the presets that reference them. There are two good options:

Save, or re-save, your preset using the 'Save Consolidated' command in the FILE menu. This places a copy of all non-Factory wav/aiff/sfz file alongside the .acp file. Then you can just submit your preset bank folder, and all of the audio data (.aaz files and samples) will be included.

Create a folder with your name (or the same name as your new preset bank) in Alchemy/Samples/User, and keep all your custom samples in it. Then submit your custom sample folder — limited to the samples you've actually used — in addition to your preset bank folder. This is the recommended procedure when more than a few custom samples are involved.

Tune presets carefully

Please ensure all presets are tuned to the usual standard (A3 = 440Hz). The exceptions are unpitched sounds such as some drums, percussion instruments, and sound effects. If your preset uses a Master Tuning other than the Default tuning, then ensure it is tuned the way users who are familiar with the tuning scheme will expect. (Normally, middle C or some other reference pitch should be tuned to the usual standard.)

Presets with no pitch-tracking, such as drum loops, should still be programmed to respond in a useful way when different notes are played across the keyboard.

Use sensible octave transpositions

Bass presets, for instance, normally sound one or two octaves lower than the default tuning. The issue is more critical when the intended range of a sound is less obvious. Users will know that a bass preset is intended to play low notes, but they shouldn't be expected to know that a sound effect needs to be played in an unusually low or high range — so adjust the Master Coarse Tune parameter to position the 'best' range in the middle of the keyboard.

Keep CPU usage as low as possible

Please ensure the CPU usage of your presets is in line with that of the Factory presets, which have been optimised not to exceed 50% CPU on a 2ghz dual Core machine when playing 6 notes. Please familiarise yourself with the CPU levels of the Factory content and ensure your presets perform comparably. The Troubleshooting page offers tips on

managing CPU load.

A preset should not reference more than 100MB of sample/aaz data

Alchemy does not perform disk streaming, so a preset using more than 100MB of data will take too long to load. Generally you shouldn't have no worry about this as almost all of the factory SFZs reference a maximum of 15MB of samples.

27.2 Sample Content Guidelines

In brief Sounds should be recorded with as little reverb and effects as possible. Files should be recorded in uncompressed WAV format at 16bit, 44kHz. Files should be named to reflect their root note. The format for this naming is Instname-Patchname-D#4.wav. 'D#4' is the name of the root note. C3 is middle C in our terminology.

In detail If a source is essentially mono, then you should probably record in mono. For instance, a flute should be recorded in a mono (because it is essentially a mono source), whereas a piano or string section should be recorded in stereo (because they are essentially stereo sources). The sounds should be recorded with as little reverb and effects as possible. Files should be recorded in uncompressed WAV format at 16bit, 44.1kHz.

Files should be named to reflect their root note (and velocity, if more than one velocity layer is to be recorded). The format for this naming is Instname-Patchname-100-C4.wav. 'C4' is the name of the root note, and '100' is the maximum velocity associated with this sample (always a value in the range 1–127). Provided this format is used, then a range of samples may be selected and automatically mapped across the keyboard. If you only provide samples in one velocity layer, then the velocity value (e.g. '-100') should be left out of the name.

Try to choose a meaningful name for the file — typically this should be the name of the instrument or synth, followed by the name of the particular playing style or patch — e.g. Cello-Bowed, PadSynth-Angels. Sample filenames should not include any protected company or brand names. Instead, choose a name which is similar, so that knowledgeable users will recognise the reference — e.g. Kog instead of Korg, or Yuno instead of Juno. Please capitalise the first letter of each word.

For acoustic instrument samples, if the instrument is capable of multiple playing styles (vibrato/no vibrato, plucked/bowed), please capture the instrument at the same level of detail for each style.

Files should have their preceding and trailing silences removed and should start and end at a zero crossing. Don't crop the audio before the sound has decayed into the noise floor. If a sound should loop, then loop points should be defined, taking care to ensure no clicking or other artefacts. Please do not shorten samples by cropping after the loop points. If you use cross-fade looping, please ensure that post loop crossfading is also applied, so that if the sample is played back with looping turned off, there are no clicks at the loop end point.

27.2.1 Sample Content — Special Cases

Drum Kits Although these are just guidelines, and we appreciate drum-kit layouts may vary depending on the sample content, drum kits should ideally be laid out as follows, starting at MIDI note 36 (two octaves below middle C):

Kick1, Kick2, Kick3, Kick4, Snare1, Snare2, Snare3, Snare4, Hat1, Hat2, Hat3, Hat4, Perc1, Perc2, Perc3, Perc4, Cymbal1, Cymbal2, Cymbal3, Cymbal4, Tom1, Tom2, Tom3, Tom4, and so on in groups of 4 where possible.

If there are only two kick drums, then the other slots should be left blank. If the kit is formed from some other kind of sound other than actual drums — e.g. children playing in a park — then they should still be layed out in this way, picking the sounds that would work best when used as a kick in the kick slot, and similarly for the other drum types.

Tempo Synced Loops Loops should loop the entire length of the sample, and you should crop the loop to work in this way. They should also contain number of beats at the end of the filename (e.g. GenericHouseLoop-16bts.wav).

Ambiences When recording ambiences, take whatever you feel is an appropriate amount of time. Typically it makes sense to record in stereo. We recommend around 30 seconds for a typical ambience, such as a railway station. Sounds should be edited so that they loop smoothly over the entire length of the sample. There should not be any fade ins/outs of the sample, as this can easily be added with an envelope.

27.2.2 SFZ file format considerations

When constructing SFZ files, be sure to limit your use of opcodes to those supported by Alchemy. A list of supported opcodes is available on the Import page of this manual.

Due to restrictions on patch loading times, you should avoid making SFZs which use more than 100MB of samples. Samples and SFZs for a particular instrument should be in the same folder, and the folder name should be the same as the instrument.

28 Credits

Lead Programmer & Designer: Ben Gillett

Programmers: Rob Martino, John Proctor, Kelly Fitz, Magnus Jonsson, Jules Vleugels

Lead Sound Designers: Tim Conrardy, Colin ‘biomechanoid’ Fraser

Sound Designers: Arksun, Artvera, Beej, Antonio Blanca, Ian Boddy, Dangerous Bear, Richard Devine, Rory Dow, Torben Hansen, Junkie XL, Michael Kastrup, Christian Kjeldsen, Bryan ‘Xenos’ Lee, John ‘Skippy’ Lehmkuhl, Paul Nauert, Frank ‘Xenox’ Neumann, Pendle Poucher, Tasmodia

Sample Library: Artemis, Ian Boddy, Tim Conrardy, Rory Dow, Dunk, Claire Fitch, Colin ‘biomechanoid’ Fraser, Galbanum, Ben Hall, Nucleus SoundLab, Robert Rich, Brad Scherick, Scott Solida, Spe3d, Allen Strange, Neil Wakling

Graphic Interface: Bitplant

Manual: Paul Nauert, Paul Sellars, ZenPunkHippy

The design of Alchemy has been a collaborative effort. Many of the people listed above have had input into its design, varying from fundamentally shaping the nature of what Alchemy is, to suggesting small additional features.

Thanks to our customers, Magnus 'Sonic Charge' Lidström (for being a great friend when needed and all round genius), FXPansion (special thanks to Angus for all his advice over the years), Big Tick Audio Software (Rhino synthesizer), Lawrence Fritts (University of Iowa Electronic Music studios), Steve Holt (SfZed software), Richard Hotchkiss (Gvox music software), Dennis Lenz (Prodyon Robocoder software), Glen Olander (Green Oak Crystal synthesizer), Redmatica (the sample library was prepared with a custom version of KeyMap), all the other people who have helped us along the way and all our friends and family for putting up with us being glued to the screen for so long. **Thank you all — we couldn’t have done it without you!**